Apple DOS 3.3
磁碟作業系統手册

- 第一部分　使用手册
- 第二部分　程式設計手册

詹朝基
蔡源斌　譯

Apple DOS 3.3
磁碟作業系統手册

- 第一部分　使用手册
- 第二部分　程式設計手册

詹朝基
蔡源斌　譯

聯星出版社　印行

第一部分

使用手册

目　　錄

序 ………………………………………… VII

第〇章　如何使用本書 ………………………… 3

簡介DOS ………………………………… 3

使用 DOS 的基本設備 ……………………… 4

DOS 的相關手册 ………………………… 4

DOS 磁碟 ………………………………… 5

本書的編排方式 …………………………… 5

一些特殊註解 ……………………………… 7

第1章　給DOS新用戶的自習教材 ……………… 11

使用入門 ………………………………… 12

關於磁碟和磁碟機 ………………………… 13

磁碟片的保養 ……………………………… 16

插入磁碟片至磁碟機內 ……………………… 17

自磁碟機內取出磁碟片 ……………………… 19

啓動 DOS ………………………………… 20

解決開機的難題 …………………………… 22

啓動（開機）的作用 ………………………… 24

重新啓動 DOS …………………………… 25

查看磁碟的內容－CATALOG指令 ……………… 26

複製 SYSTEM MASTER 磁碟片 …… 29
一部磁碟機的拷貝過程 …… 31
兩部磁碟機的拷貝過程 …… 36
拷貝磁碟出了問題時 …… 40
測試拷貝版磁碟 …… 41
預備一空白磁碟片－ INIT 指令 …… 43
使用程式 …… 48
自磁碟中讀取資訊－LOAD指令 …… 48
改變記憶體的內含資訊 …… 49
儲存資訊至磁碟片－SAVE 指令 …… 52
執行磁碟內的程式－RUN指令 …… 54
關掉電腦 …… 55
第一章摘要 …… 56

第2章　DOS概觀 …… 61

DOS 如何動作 …… 61
監督程式 …… 62
啓動的過程 …… 62
檔案及磁碟片卷號 …… 68
檔案 …… 68
卷號 …… 70
DOS 指令 …… 71
語法 …… 71
引數 …… 75
第二章摘要 …… 78

第3章　使用磁碟 …… 83

CATALOG指令 …… 83
了解目錄的意義 …… 85
拷貝用程式 …… 87
Applesoft BASIC下的COPYA程式 …… 87
Integer BASIC下的COPY程式 …… 89
拷貝過程中的錯誤訊息 …… 89
準備好可用的磁碟片 …… 90
啓始化的磁碟 …… 91
主磁碟 …… 92
歡迎詞程式 …… 92
INIT指令 …… 93
決定擴充槽和磁碟機代號：SLOT#程式 …… 95
第3章摘要 …… 96

第4章 使用檔案 …… 99

RENAME指令 …… 99
LOCK指令 …… 102
UNLOCK指令 …… 103
DELETE指令 …… 104
VERIFY指令 …… 105
FILEM程式 …… 106
指定擴充槽編號和磁碟機代號 …… 107
利用"通配字元"指定檔案名稱 …… 108
選擇FILEM的功能項 …… 110
FILEM顯示的訊息 …… 120
第4章摘要 …… 121

第5章　使用已存於磁碟上的各種程式……125
RUN指令……125
LOAD指令……129
SAVE指令……130
與其他週邊裝置溝通……133
PR #指令……135
IN #指令……137
重新啓動 DOS 的途徑……138
結論……139
第 5 章摘要……140
附錄A　轉換13磁區磁碟的格式……145
轉換 13 磁區的磁碟：CONVERT13程式……146
實例……147
重複名稱的檔名……151
通配字元……151
不需轉換即可執行 13 磁區磁碟的程式……152
使用START13程式……152
使用BASICS 磁碟……154
附錄B　DOS錯誤訊息……157
DOS 錯誤訊息……157
DISK FULL……158
END OF DATA……159
FILE LOCKED……159
FILE NOT FOUND……159

FILE TYPE MISMATCH …… 160
I/O ERROR …… 161
LANGUAGE NOT AVAILABLE …… 162
NO BUFFERS AVAILABLE …… 163
NOT DIRECT COMMAND …… 164
PROGRAM TOO LARGE …… 164
RANGE ERROR …… 165
SYNTAX ERROR …… 166
VOLUME MISMATCH …… 166
WRITE PROTECTED …… 166
停止程式的執行 …… 167

附錄C　DOS磁碟內的各種程式 …… 169

SYSTEM MASTER 磁碟內的程式 …… 169
SAMPLE PROGRAMS 磁碟內的程式 …… 171

附錄D　DOS操作概念及指令摘要 …… 175

DOS 操作概念 …… 175
啓動 …… 175
重新啓動 DOS …… 176
磁碟容量 …… 177
記號說明 …… 177
語法 …… 178
引數 …… 178
指令摘要 …… 179
CATALOG …… 180
DELETE …… 181

IN# 181
INIT 181
LOAD 182
LOCK 183
PR# 183
RENAME 184
RUN 184
SAVE 184
UNLOCK 185
VERIFY 185

序

DOS 程式設計者手册（ DOS Programmer's Manual ）係供欲將程式和其他資訊寫入磁碟者參考之用，其內容描述 DOS 3.3磁碟作業系統（ disk operating system），使您很容易將資訊放入磁碟片，修改磁碟上的資訊，擷取磁碟上的資訊。DOS 適用於 Apple II,Apple II Plus 和Apple IIe 電腦。

DOS 使用手册（ DOS User's Manual ）可做爲本手册的簡介，不過本手册也摘錄其相關的資訊。

本書的架構

本書的設計供參考之用，其涵蓋的內容由簡而繁。

各章以主題簡介開頭，並以討論過的指令摘要作結束。

第 1 章，描述 DOS 的使用環境和主要係自鍵盤下達的DOS 指令。

第 2 章，解說使用 DOS 指令的幾種方式。它將說明如何於程式中含有 DOS 指令，如何監督程式的執行過程，以及如何攔截錯誤作適當的處理。

第 3 章，告訴您有關於文字檔的磁碟存取方式，並比較循序和隨意存取的同異點。本章主要是介紹如何設計有關於循序存取文字檔的 BASIC 程式。

第 4 章，解說隨意存取檔的種種，以及如何撰寫程式來處理隨意存取文字檔。

第 5 章，告訴您如何使用 DOS 去設計可執行其他程式的程式，以及如何連接（ chain ）程式，和創製“ 轉鑰式 ”（ turnkey ）程式的方法。

第 6 章，描述有關於二進制機器碼檔案（binary file）的指令。它同時也討論 DOS 和監督程式（monitor）之間的關係。

附錄A，指導您如何在 13 磁區格式的磁碟中執行程式，如何將 13 磁區格式的磁碟轉換成 16 磁區磁碟。

附錄B，解說磁碟的儲存過程：磁軌和磁區如何付予配置圖以了解 DOS 使用它們的情形，機器語言設計師如何直接指明磁軌和磁區的位址。

附錄C，是 DOS 錯誤訊息的詳說及其解決之道。

附錄D，說明存於SYSTEM MASTER和SAMPLE PROGRAMS 磁碟內各程式的作用。

附錄E，是所有 DOS 指令的摘要整理。

最後的"名詞解釋"一節闡示本書所使用到的各術語之意義。這些術語在本書第一次出現時曾以粗體字加以標記的。同時，也特別附上有關的：常用的 DOS 指令之"快速查閱表"（速見表），讓您能在需用的時候立即獲知操作大綱和簡要的說明。

0 如何使用本書

■簡介DOS 3
- 使用DOS的基本設備 4
- DOS的相關手册 4
- DOS磁碟 5

■本書的編排方式 5
■一些特殊註解 7

第0章 如何使用本書

簡介 DOS

磁碟作業系統（ disk operating system ）是一組電腦程式，負責處理磁碟片的資訊，它包含了儲存資訊至磁碟、讀回資訊傳至電腦的存取動作。

DOS是 Apple II 電腦的特定磁碟作業系統，它是各種資訊的“管理者”。此處的“資訊”（ information ）可能是可執行特殊工作的“程式”（ program），或是名稱、數目等合聚而成的“資料”（ data ）。但是，DOS 並無法為您服務下列類似的事項：幫您撰寫一商業書信，或庫存的現況記錄，或繪製減肥過程的曲綫圖。DOS 的性質是在建立資訊（程式或資料）之後，讓您能儲存和管理這一類的資訊。

DOS 3.3 將磁碟的表面規劃成 16 個磁區（ sector ）。假若您想在 16 磁區格式的DOS系統下，使用較早期的 13 磁區格式之磁碟，請詳讀附錄A 。

使用DOS 的基本設備

在繼續本書之前，請先安裝好您的Apple II電腦，可按照附在電腦的簡要安裝手册內說明，安裝您的電腦系統。

要使用 DOS ，您的 Apple II 電腦至少須有 **16K位元組的"隨意存取記憶體"** （ 16 kilobytes ，K是代表“仟”，然而K的眞正意義是1024；位元組，Byte ，是記憶體的一種基本單位，由八個位元（ bit ）構成，一個位元組的內含值介於0和255之間。隨意存取記憶體， random-access memory ，簡稱RAM ，是可反覆地讀出和寫入資訊的一種記憶體。）DOS 本身的程式約佔用10.5K的記憶體，希望提供足夠的記憶空間給DOS和 **應用程式**（ application program ），我們建議您使用48K的系統。（註：“應用程式”是運用電腦的資源和能力以執行特定用途的程式，例如文書處理程式即屬於此類。而 DOS 本身的程式是稱爲“ **系統程式** ”（ system program ），它提供一些基本的、一般性的計算、管理資訊的能力，而應用程式即透過它與電腦本身硬體或軟體溝通，執行各種特定用途的工作。）

您的Apple II電腦至少須要有一部 **磁碟機**（ disk drive ）。至於該如何將磁碟機接上 Apple II 主機， 請參考磁碟機的使用、安裝說明。

DOS 的相關手册

DOS 使用者手册（ DOS User's Manual ，即本中文譯書的上半部）是初步的入門書，一步一步地詳細解說使用 DOS 的基本原理。假若您正在學習程式設計或想執行已寫好的程式或遊戲程式，本手册可以提供基本而必要的知識。在您讀完附在Apple II電腦

的簡介性手册說明之後，您就可以準備翻閱本書。

DOS程式設計者手册（ DOS Programmer's Manual ，即本中文譯書的下半部）是進一步深入解說的參考書，它會說明較複雜的 DOS 指令用法，以及如何配合BASIC 程式使用各種指令。假若您對程式設計略有經驗並讀過較基本的使用手册，您就可以準備閱讀此書。

DOS 磁碟

在標籤上註明 DOS 3.3 SYSTEM MASTER 的磁碟片含有管理磁碟及檔案的 DOS 程式。當您使用了 SYSTEM MASTER ，DOS 的各種指令及相關能力會擴增原先Apple II主機提供的能力，讓您更能得心應手管理資訊。

註明 DOS 3.3 SAMPLE PROGRAMS 的磁碟片含有一些示範用程式，在本書中會使用它們來向您闡示 DOS 指令的用法。同時，此 SAMPLE PROGRAMS 含有一些 BASIC程式例子，對懂得 BASIC 程式的人而言，可能會有興趣看看它們的。

本書的編排方式

本書適合剛開始學習電腦的人——準備執行已存於磁碟的程式，或者想設計簡單的 BASIC 程式的一般人。

第 1 章“給 DOS 新用戶的自習教材”是特別爲初次使用電腦的人編寫的一章。假若您已對電腦的應用有些經驗，可以自第 2 章開始讀起，只需在必要的時候才去翻閱第 1 章的部份章節。

每一章的開頭均先簡短的介紹該章的內容，接著章節中的例子可引導您逐步學習使用DOS，試著依循書上的操作步驟，對您的學習會有很大的益處。在每一章後頭有個摘要，複習該章的指令意義

及操作觀念。

下列是“DOS 使用者手册”內各章的概要簡說：

- 第 1 章，“給 DOS 新用戶的自習教材”，讓一般讀者在數小時學習使用 DOS 的基本用法。本章會特別告訴您，如何插入磁碟至磁碟機內，如何啓動電腦，如何複製磁碟以預防無意中毀損後尚有備份拷貝，如何準備好一磁碟片以接收資訊，如何使用磁碟的檔案，以及如何關掉電腦。

  同時，本章也介紹一些基本的術語、電腦觀念和操作技巧。包含了檔案是什麼，如何載入和執行一程式，如何儲存一檔案，如何處理意外的事項。
- 第 2 章，“DOS 概觀”，告訴您使用 DOS 之前的基本知識，說明檔案的種種以及如何命名它們，並告訴您使用 DOS 的基本法則，及書中各種記號的代表意義。
- 第 3 章，“使用磁碟”，將磁碟當作一完整的個體——查看磁碟內已存入什麼資訊，複製整片磁碟，以及預備好一可用的新磁碟以接受資訊。
- 第 4 章，“使用檔案”，則個別處理各個檔案：追踪記錄目前的狀況，更動檔案名稱，加上保護措施以預防無意中的毀損，必要時删除某些檔案。
- 第 5 章，“使用已存於磁碟的各種程式”，討論有關使用應用程式的事項：如何執行它們，如何保存它們輸出的中間結果。本章同時也含有 DOS 如何與其他週邊裝置（如印表機）溝通的方式。
- 附錄 A，“轉換 13 磁區磁碟的格式”，指引擁有先前版本 13 磁區磁碟的人，如何在目前 16 磁區的DOS 3.3系統下使用 13 磁區的磁碟，如何轉換成 16 磁區格式的磁碟。
- 附錄 B，“DOS 錯誤訊息”，討論出現警告和錯誤訊息的來

源，此處列舉來自 DOS 的錯誤訊息，逐條解說出錯誤的原因並建議如何修正錯誤。

- 附錄C，“ DOS 磁碟內的各種程式 ”，簡要地描述在 SYSTEM MASTER 和 SAMPLE PROGRAMS 磁碟內的各種程式的作用。
- 附錄D，“ DOS 操作概要及指令摘要 ”爲 DOS 指令作個簡要的說明，您可以查閱本附錄很快地獲知各指令的用法。
- “ 名詞解釋 ”一章裡定義您一些可能不甚熟悉的電腦術語，在本書第一次出現的章節會以“ 粗體字 ”加以標示的。

（譯註：本中文譯書的“ 名詞解釋 ”一章與 DOS 程式設計者手册的“ 名詞解釋 ”合併。附在本中文書的後頭，內容更充實可觀；惟顧及不同階層人士的電腦知識背景，譯者於適當的章節內加入“ 術語小註 ”，擇要解說各名詞的意義。）

一些特殊註解

在本書提到的“ Apple II ”是泛指 Apple II 的各種機型：Apple II，Apple II Plus 以及 Apple IIe 。當某種操作方式僅適用某種機型，會在文中特別提出機型說明。

重要的電腦術語第一次出現於文章時，以粗體字標示，同時在“ 名詞解釋 ”也可找到完整的定義。

本書中有各種字體，相當於不同的視覺效果，例如下列方整的字體，表示顯示於螢幕的訊息，

```
IT LOOKS LIKE THIS.
```

；假若您應該忠實地按下鍵盤的某些字鍵，則以下列方式表示之，

[R][U][N][SPACE][C][O][L][O][R][SPACE][T][E][S][T]

雖然在鍵盤上的空間棒並未標示 [SPACE] ，但它是提醒您應該打入一個空白字元，所以視“空白鍵”爲一個字鍵。假若您應打入的一個字鍵，雖以兩個字元標示您仍是打入文中標示的那一個字元。[SHIFT] 的記號即是代替按此鍵以得到希望的字元。

如果您看到兩個字鍵以“橫劃”（hyphen）加以連接，表示您必須同時按下兩個字鍵，例如 [CONTROL]-[RESET] 即代表須同時按下 [CONTROL] 和 [RESET] 鍵。實際上的操作裡，您可以先按下 [CONTROL] 鍵，暫時不改掉它，而後再按下 [RESET] ，有相同的效果。

本書中各章節可能出現一些小方塊文章，以“警告”加以標示，請特別留意。同時章節內也可能夾雜“順便一提：”（By the Way：），“註：”（Note：）的小方塊，可以提供精要的特別註解，幫助您了解文章的內容。

1 給DOS新用戶的自習教材

■使用入門　12

■關於磁碟和磁碟機　13

• 磁碟片的保養　16
• 插入磁碟片至磁碟機內　17
• 自磁碟機內取出磁碟片　19

■啓動DOS　20

• 解決開機的難題　22
  - △指示燈亮了或不亮　22
  - △空白畫面　22
  - △沒有任何訊息或出現奇異的訊息　23
  - △奇怪的聲音　23
• 啓動（開機）的作用　24
• 重新啓動DOS　25

■查看磁碟的內容——CATALOG指令　26

■複製SYSTEM MASTER磁碟片　29

• 一部磁碟機的拷貝過程　31
• 兩部磁碟機的拷貝過程　36

■拷貝磁碟出了問題時　40

• 測試拷貝版磁碟　41

■預備一空白磁碟片——INIT指令　43

■使用程式　48

• 自磁碟中讀取資訊——LOAD指令　48
• 改變記憶體的內含資訊　49

- 儲存資訊至磁碟片——SAVE 指令 52
- 執行磁碟内的程式——RUN 指令 54

■關掉電腦 55

■第 1 章摘要 56

第1章

給DOS新用戶的自習教材

本章簡介您可能一而再地使用的各種 DOS 指令。同時，說明什麼是磁碟，以及如何保存（照顧）它們；檔案是什麼，執行一程式或儲存一程式該如何做？本章並附帶說明若有意外事項，該如何處理它。

本章的編排方式有點像教材，每一重點均先提出該重點工作的概觀，而後再逐步地解說各步驟。

請從頭開始閱讀，並試著嘗試每一操作步驟，順著章節連續地練習，到了本章的後頭，您將學會：

- 如何保存（保養）磁碟。
- 如何將一磁碟插入磁碟機內。
- 如何啓動電腦。
- 如何複製整片磁碟，保存一備份磁碟。
- 如何準備好一磁碟以備接受（儲存）資訊。
- 如何自磁碟中取得資訊。
- 如何存入資訊至磁碟內。
- 如何執行一程式。
- 如何關閉電腦。

假若您尚未翻閱 Apple II 電腦附送的簡介性手册或操作須知

，現在請先花點時間讀完這些說明。

順便一提：在序文曾提起過的“一些特殊註解”，會說明一些特殊的記號，例如“警告方塊”會陸續在文中出現，請讀者詳讀內容的說明文字。

使用入門

“學習使用DOS”意味著學習一些特殊指令，例如 CATALOG 指令，磁碟片（ disk ）插入或取出磁碟機（ disk drive ）的操作方法，以及一些新的字語（術語），如“載入”（ load ），“執行”（ run ）。這些字語並不會讓您覺得迷惑難解，祇是一些平常的、可以理解的技術性用語。

DOS 可以幫助您“控制”磁碟片內含的資訊。“控制”是表示您可以儲存資訊至磁碟上，而後又能讀回資訊傳送給電腦。什麼是“資訊”（ information ）？這就看您了！它可能是可完成一件特殊工作的 **程式**（PROGRAM），或者一串名稱、數值合聚而成的資料，甚至是任何您想像、希望的東西。您不妨將“資訊”視爲有某種意義的資料之泛稱或代名詞。

當您想依循本教材的指引來學習 DOS 時，您應該預備好下列設備或磁碟：

- 一台 Apple II 電腦，並配備一台或二台磁碟機。
- DOS 3.3 SYSTEM MASTER 磁碟。
- DOS 3.3 SAMPLE PROGRAMS 磁碟。
- 至少二片空白磁碟。

DOS 3.3 SYSTEM MASTER 磁碟是很特殊的，它含有構成 DOS 的一些程式。在本教材的課程安排裏，您會一直使用此

磁碟片。至於 SAMPLE PROGRAMS 磁碟，則內含各種示範程式及遊戲程式，本章將會先介紹其中的一個程式。

電腦術語小註：

DOS——Disk Operating System（磁碟作業系統）之簡稱，負責管理磁碟內含資訊的各種存取工作。

PROGRAM——程式，一組指令，指示電腦如何從事某些工作。

關於磁碟和磁碟機

一部**磁碟機**（disk drive）的體積比鞋盒小很多。假若您可以卸下外蓋透視內部，您將看到具有二種機械裝置的磁性儲存裝置。一個機構讓磁碟片運轉，另一個機構是在磁碟片的表面讀取和寫入資訊。

當磁碟機在磁碟片表面"接收"資訊，它的功能就有如錄音機，然而接收自磁碟的資訊是要傳送給電腦，並不是一般的擴音器。磁碟機也可以記錄，或稱"寫入"，電腦傳送給磁碟的資訊。

一片"**軟性磁碟機**"（flexible disk）其外觀是直徑5英吋左右的平滑、有彈性的圓形薄片（一般是用方形塑膠框加以包裝的），在中心處有個圓孔。有時候，"flexible disk"也稱爲"floppy disk"（其中譯均爲"軟性磁碟"）。

磁碟表面敷有一層磁性物質並加上外包裝以備儲存資訊，其道理就如錄音帶外加盒子一樣，當您拿到一磁碟片，須適當地保護磁碟，以防損害。

正常情況下，您只能看到磁碟表面的一小部份，因爲磁碟片是以黑色、方形的塑膠護套（plastic jacket）包裝起來的。此外套

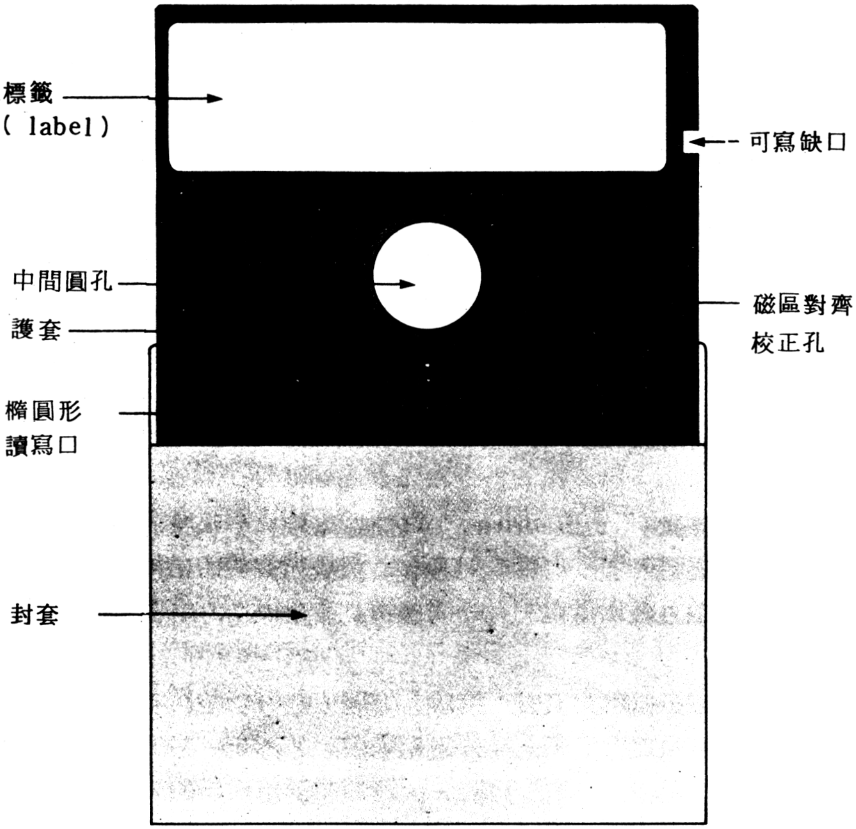


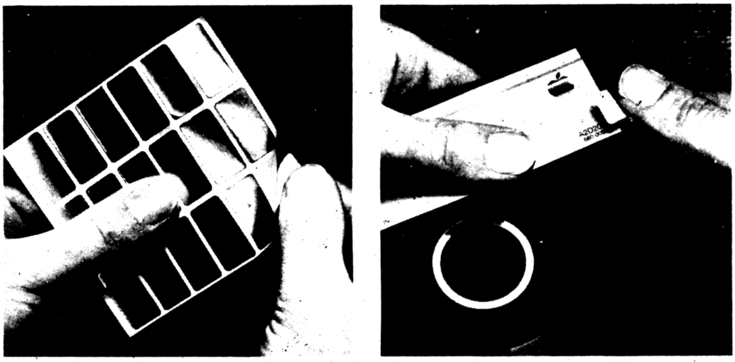
圖 1-1　磁碟的各部位名稱

兼具潤滑和保護磁碟的作用，在其上有兩個切口（洞），其中一個是橢圓形的切口（長約一吋），是磁碟片被讀取或寫入資訊的“出入口”（window）。較小的方形缺口是"**可寫缺口**"（**write-enable notch**），可以讓您改變磁碟上的內含資訊，假若沒有該缺口，則電腦僅能讀出磁碟片的資訊，無法再寫入資訊至磁碟。

某些磁碟片（例如隨電腦附送的 **SYSTEM MASTER** 磁碟）

並沒有“可寫缺口”，其作用是保證可以保護儲存其上的重要資訊。

假若您想預防磁碟上的資訊無意中被毀損，請在可寫缺口貼上 "**寫入保護貼紙**"（ Write-protect tab ）。圖 1-2 是爲磁碟片加上寫入保護貼紙的步驟解說圖。



1　　2

圖 1-2　爲磁碟片加上寫入保護貼紙，標示的數目對應至文章的解說步驟。

1. 請拿出跟隨磁碟片附送的有黏性、銀質貼紙。請取出其中一張貼紙。
2. 將貼紙的前半部份黏在磁碟外套上，覆蓋在可寫缺口，並折疊另後半部份至外套背面。貼完後，請檢查是否貼得平順。

任何時候，您想增添一些資訊至已加上寫入保護的磁碟，只要再去掉該貼紙便可以了。

磁碟片的保養

下列是照顧好磁碟片的簡單規則，圖1-3可以告訴您一些簡要事項。

- 請用手握住磁碟片外套及標籤的部份。千萬不要自塑膠外套取出磁碟片，或者觸摸到磁碟的表面。尤其注意到在標籤紙寫字時，不要碰到磁碟片上橢圓形的讀寫部份。

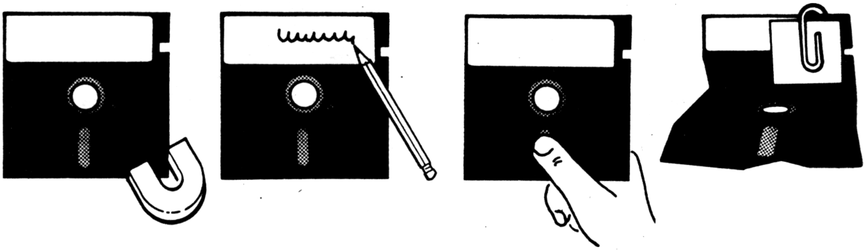


圖1-3　四種不正當的磁碟保養方式

- 在貼有標籤的磁碟加註文字時，請用平滑筆頭書寫之。不要重壓磁碟片表面。最好是先寫好標籤，才貼到磁碟上。
- 不要用鉛筆或圓球筆（墨水筆）在已貼有標籤的磁碟上書寫文字，因爲可能在磁碟片表面造成筆屑之類的污損。
- 不要在標籤上使用橡皮擦。可能造成磁碟表面的污損。
- 不使用磁碟片時，請收放在紙封套內。
- 正常地收藏磁碟片於紙封套內，不要折疊，或夾著迴紋針之類的東西。
- 磁碟片的保存地點應遠離陽光的照射，濕氣的侵襲以及高

、低溫的環境。

- 不要使磁碟片接近磁場、電子裝置（尤其是電視機和大型馬達。暫時性擺放於電腦或磁碟機的旁邊是可以的。）

插入磁碟片至磁碟機內

- 如圖 1-4 所示，輕輕地打開第一部磁碟機的門。假若在磁碟機內有任何磁碟片，請小心地取出並放入紙封套內。

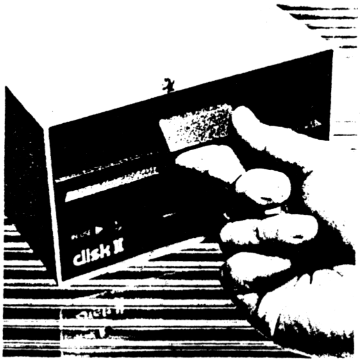


圖 1-4　打開磁碟機的門

- 自紙封套內，取出 DOS 3.3 SYSTEM MASTER 磁碟片（圖 1-5）。
- 用大姆指握住磁碟片的標籤部份，輕輕地水平推入磁碟機的溝槽內。磁碟片應是橢圓形讀寫口先進入磁碟機內，最後是標籤的部份。注意不要彎曲或強力推入磁碟。假若覺得有阻力不甚順利，慢慢地拉出磁碟片，再試一次。

假若該磁碟並無標籤，則拿住磁碟時，可寫缺口應是在“左方”。

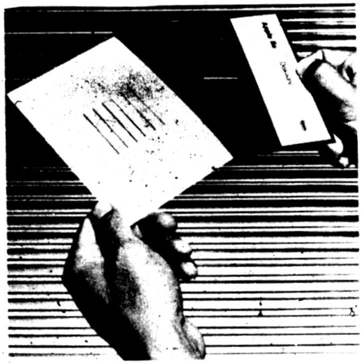


圖 1-5　自封套內取出磁碟片

圖 1-6　插入磁碟片

- 當磁碟片已全推入磁碟機內，將磁碟機的門拉下，您會聽到一聲門卡上的聲音。

圖 1-7　關好磁碟機的門

自磁碟機內取出磁碟片

當磁碟機的指示燈亮著，即表示 DOS 正在讀取或寫入資訊。自磁碟機內取出任何磁碟片之前，應先確定磁碟機的指示燈已滅，否則可能毀損磁碟片的內含資訊。打開磁碟機的門，輕輕地拉出磁碟片，試試看！

1. 打開磁碟機的門。
2. 輕輕地自磁碟機內取出 SYSTEM MASTER 磁碟。

警告：當磁碟機的指示燈還亮著的時候，切勿取出磁碟片，因為可能危害或誤讀、誤寫磁碟資訊。有時候，磁碟機一直在空轉，您可以按下 CONTROL − RESET 移動位置。

啓動 DOS

要使用 DOS ，您必須將電腦主機開關打開，將磁碟片上的 DOS 讀入 **主記憶體**（main memory）。將 DOS 讀入記憶體，其意義是將含有 DOS 的程式複製一份，存入主機內的記憶體以啓動電腦。整個過程如下所示：

1. 先打開 **顯示裝置**（ display device ） 的開關，它可能是顯示器（ video monitor ）或家用電視（將聲音調小）；這是啓動前的準備工夫。
2. 查驗 SYSTEM MASTER 磁碟片是否已放入第一部磁碟機，並將磁碟機的門關好。
3. 將手伸到電腦主機的左後面，那裏有個電源開關（如圖 1-8 所示），請將它按下，開啓開關。

圖 1-8　打開主機的電源開關

此時，鍵盤上的電源指示燈亮了，同時第一部磁碟機上的燈也會亮，並在讀入磁碟時，有些嘈雜的運轉聲音。稍候數秒後，不再有聲

音且磁碟機的指示燈熄滅了。假若您的情況不是如上所述，請閱讀下一節"解決開機的難題"，以了解發生了什麼事。

不管您的主機連接了幾部磁碟機，務必將SYSTEM MASTER磁碟片插入第一部磁碟機（ drive 1 ）才能啓動 **DOS** 工作。

一旦您成功地啓動 **DOS** ，顯示裝置的螢幕將會顯示如圖1-9的畫面，它告訴您 **DOS** 已存入記憶體，並準備接受您的指令（ Command ）。假若您沒看到如下圖的畫面，請查看下一章節的內容，解決這一項問題。

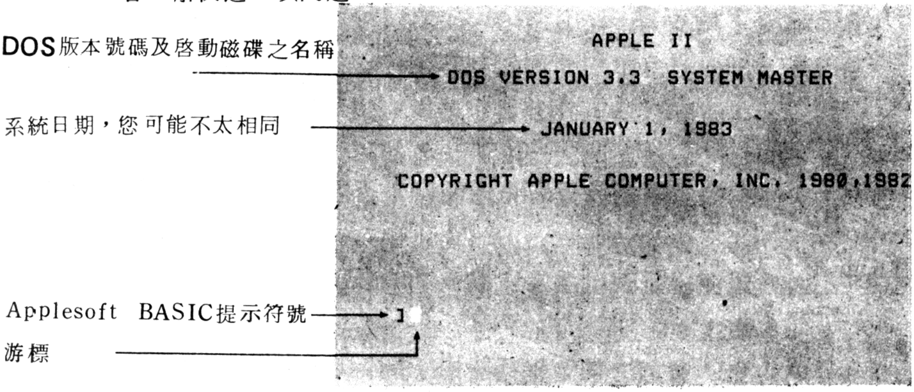


圖 1-9　SYSTEM MASTER啓動後的螢幕畫面

按鍵事項：假若您的主機是屬於 Apple IIe型，在啓動 DOS 時將多顯示一行訊息BE SURE CAPS LOCK IS DOWN，請您務必按下 CAPS LOCK 鍵。若屬於 Apple II 或 Apple II Plus 機型，不會顯示該段訊息，您不用操心 CAPS LOCK 鍵的問題。

電腦術語小註：

Main memory——主記憶體，它是 Apple II 電腦系統內的一部份，其作用是儲存您的各種資訊（資料或程式）。

Display device ——顯示裝置，將各種資訊呈現在您面前的裝置。

解決開機的難題

假如開機的程序無法正常工作，本節將探討出錯的原因，其實這一類的問題大多很容易解決。

指示燈亮了或不亮

倘若主機的電源指示燈或顯示裝置的指示器不亮，請遵循下列步驟查看原因：

- 主機和顯示裝置的電源開關是否已打開（放在 ON 位置）？
- 電源線是否已插入主機後面的插座？可能鬆掉了，檢查看看！
- 會不會您的電源線插座出了問題？用收音機之類的電子產品測試插座是否完好。

空白畫面

假若螢幕是一片空白，查驗下列事項：

- 顯示器的信號線（ Cable ）其兩端是否已分別插入顯示器及電腦主機的插座。
- 顯示器的“亮度鈕”是否已調整好了？或者試試“對比鈕”？

沒有任何訊息或出現奇異的訊息

假若在顯示螢幕上看不到有關 DOS 的訊息（也許祇顯示 Apple II 或整個不同的訊息），表示電腦無法正確地將 DOS 自磁碟片讀出並存入記憶體內。

- 確定磁碟機是否已接上主機。
- 查驗 SYSTEM MASTER 磁碟片是否插在第一部磁碟機。
- 如果您接上了兩部磁碟機，而沒有磁碟片在內的一部磁碟機在空轉（指示燈一直亮著而且有運轉聲音），表示您將標示 drive 1 的貼紙弄錯了。請先關掉電腦，而後將標示更正，或抽換磁碟片，將 SYSTEM MASTER 磁碟自第二部磁碟機抽出，插入眞正的第一部磁碟機。
- 若螢幕上顯示的是個星號（ * ），請打入

  [6][CONTROL]-[P]

  以啓動DOS。

  假若打開Apple電腦，最先看到的是星號（ * ），表示目前是在“監督程式”（ monitor program ）狀態，可能是您的主機並無自動啓動的 ROM（唯讀記憶體）。此時，您祇需打入 [6][CONTROL]-[P] 就可啓動 DOS 。關於自動啓動的 ROM，請詢問您的經銷商。

奇怪的聲音

假若磁碟機不停地空轉，夾雜著尖銳的碰撞聲，不妨將磁碟機的門先打開再關上。讓它的運轉聲音較有規律些。假若您的調整有效的話，磁碟機讀出磁碟內容時，將會是另一種平順的聲音。

倘若磁碟機仍在空轉，可按下 CONTROL - RESET (兩鍵同時按下）以停止磁碟機。（請讀下列的“**警告**”事項）當磁碟機的指示燈熄滅了，將磁碟片取出，再放入磁碟機內。務必平穩地推入磁碟機中，不要將它彎曲或斜放插入磁碟機。

警告：不要在磁碟機的指示燈亮著的時候，強行抽出磁碟片，因爲您可能會危害到磁碟片的內容。要停止運轉中的磁碟機，請同時按下 CONTROL 和 RESET 鍵。

一旦您已遵循上述的建議解決您的問題，請再回至“啓動DOS”一節，重新正確的開機手續。

啓動（開機）的作用

現在，您已經完成啓動DOS 的工作，或許您會很好奇的詢問“啓動 DOS 的目的爲何？它又作了什麼事？”。當您啓動 DOS，其效用是賦予 Apple II 電腦接受各種 DOS 指令的能力，及結合 DOS 本身功能的資料運算處理能力。啓動它，即是將各種DOS指令自 SYSTEM MASTER 磁碟片讀出並存入記憶體。

一旦已存入記憶體， DOS 會啓動BASIC ，而 BASIC 會在螢幕上印出一個特殊含意的**提示字元**（ prompt character ），告知您電腦已經準備好接受您鍵入的指令。若您看到的是個右方括弧（]），表示目前是在Applesoft BASIC 運轉狀態下，倘若該提示字元是大於符號（ > ），則表示處於 Integer BASIC 狀態下。

在提示字元右方的是個閃動的**游標**（ cursor ）。這個閃動的游標告訴您，您打入指令的位置或是您打入的文句在螢幕的“下一個”顯示位置。游標的樣式可能是個方格子、長方形或是棋盤交叉的方塊圖案，也許它僅是一條横線。可能它有規律地閃爍著或變化外形的尺寸。它完全是看您所使用的 Apple II 機型，以及您的顯

示螢幕的文字寬度特性而定。

重新啓動 DOS

有好幾種可能性會讓您想重新啓動 DOS ，例如，您的程式已弄亂了，已無法使用 DOS 指令也看不到提示字元，或者顯示器的螢幕出現了奇怪的圖案，磁碟機不知原因的空轉著。

假若電腦主機的電源開關仍開啓著，可以同時按下 CONTROL 和 RESET 鍵，再放掉這兩個字鍵，就可重新啓動 DOS 。一旦您是使用 CONTROL-RESET 鍵啓動 DOS ，祇顯示一個提示字元的符號於螢幕上，告訴您電腦已經可以接受 DOS 指令。立即動手試試看！您會聽到一聲"嗶"，而後顯示 BASIC 提示字元，表示您確實再啓動了 DOS 。

重新啓動，亦即是中斷電腦目前的工作，返回最初的狀態。它可節省您時間，同時也延長電腦主機的壽命（因爲不必反覆地扳動電源開關）。假若您的程式已存在於記憶體內，利用本法重啓DOS，不會漏失您的程式，但若是關掉開關再重開，則大部份狀況下會漏失已在記憶體內的資料。

電腦術語小註：

Read－Only Memory, ROM——唯讀記憶體，意即僅能讀取而不能改變在生產過程中已寫入的資訊的一種記憶體。儲存於內的資訊將永久地保留下來，您已無法加以消除或改變。

Cursor ——游標，一種標記物，告訴您下一個動作或從鍵盤上打入文字（字元）在螢幕上顯示的下一個位置。

查看磁碟的內容— CATALOG 指令

一旦您已啓動了 DOS ，您就可以查看磁碟片的“內容”(contents)，它是記載在磁碟片的目錄(catalog) 部份。此目錄其實是一長列的檔案名稱和有關各檔案的性質簡說。下列是 CATALOG 指令查看磁碟片目錄的過程：

(**關於"空白"的說明** ：在您開始打入指令之前和正式使用 DOS 之前，您有必要知道電腦是如何瞭解您打入的字句。電腦是完全忠實地接收您打入的每一個字元，也包括“空白字元”(space character) 在內。“空白”也是一種“字元”。假若您需要的是有個“空白”，您必須按下 SPACE 鍵。若僅是用方向鍵移動游標，則不會在字句之間插入空白字元。)

操作程序		**結果**
1.	確定 SYSTEM MASTER 磁碟片已插在第一部磁碟機，且DOS已啓動。	在螢幕上有個 BASIC 提示字元。
2.	打入 C A T A L O G 看看螢幕顯示的是否即是您剛打入的指令名稱。在您按下 RETURN 鍵之前，可以利用 ← 鍵將游標倒退回去修改錯誤的地方，然後再打入該指令。	假若指令名稱拼錯了字，在按下 RETURN 鍵後，會聽到“嗶”一聲，並顯示出 SYNTAX ERROR 一列錯誤訊息。沒有造成任何損害。只要再重新打入正確的指令名稱，並檢查拼字後才按下 RETURN 鍵即可。
	只在確定打入的是正確的指令才按下 RETURN 。	稍後數秒，您在螢幕上會看到 SYSTEM MASTER 磁碟片的

目錄。

（**註**：所有 DOS 和 BASIC 指令均需以大寫字母表示之。假若您的主機是屬於 Apple IIe，請先按下 [CAPS LOCK] 才能有大寫字母的效果。（但，對於標示兩種意義的字鍵，您仍然得同時按住 [SHIFT] 鍵和該雙意鍵，例如 [%/5] 才能得到標示於上面的字元。）倘若顯示的是 SYNTAX ERROR（語法錯誤）訊息，極有可能是打入小寫字母的指令。請按下 [CAPS LOCK] 聽到鍵盤一聲輕脆的聲音後，重新打入正確的指令。）

3.	按下 [SPACE] 鍵，繼續查看目錄的其餘部份。	請注視螢幕，情況應如圖 1-10 所示。

（**註**：一旦一磁碟片的目錄內容超過螢幕一個畫面所能容納的，會暫時"凍解"畫面，除非您再按下 [SPACE] 鍵，才會繼續顯示未定的部份。按 [SPACE] 鍵以繼續顯示，是 Apple 電腦的慣用法。）

---

① 第 1 欄的標示字（星號或空白）表明某**檔案**（file）是否已"**鎖住**"（lock）。有星號（*）在旁表示該檔案已鎖住，也即是在未解鎖之前您無法改變該檔案的內容。倘若第 1 欄保持空白，表示該檔案並**無鎖住**（Unlock，解鎖），您可以隨時加以更新之。

② 第 2 欄是表明**檔案型態**（file type）。在本圖中，您看到的是

- A 代表"Applesoft" BASIC
- I 代表"Integer" BASIC
- B 代表"binary"（機器語言碼）

檔案的型態告知您該檔案所屬的語言種類，以及那一類的指令才能存取（access）該檔案的內容。某些 DOS 指令僅對某種型

態的檔案有效，本書僅討論Applesoft 類和 Integer 類的檔案。讀者可在第三章了解更多的“檔案型態”細節。

③ 第 3 欄是顯示該檔案的長度(大小)。例如HELLO 檔是佔有 3 **磁區**(Sector)的空間，它是個很小的檔案，而 110磁區是頗大的檔案。(註：“磁區”是相當於 128 Byte ，將在本書第二部分的附錄B 解說它的意義。)

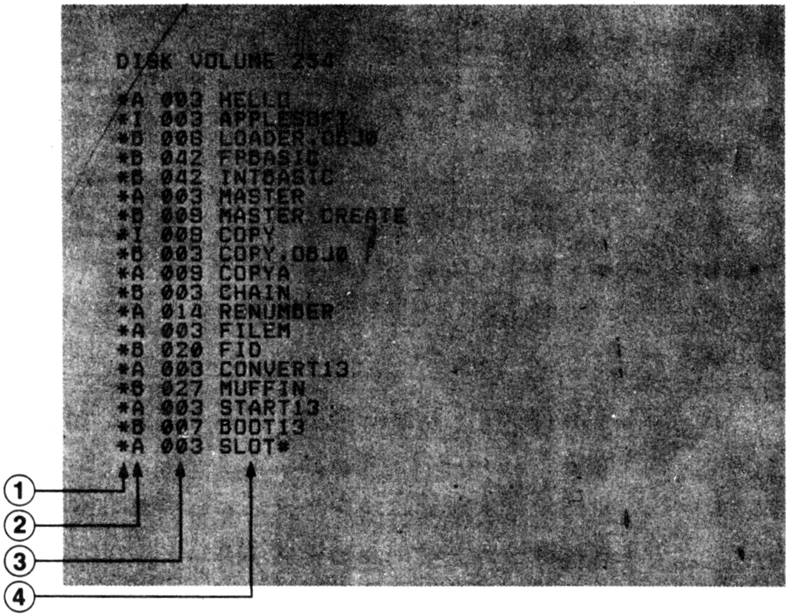


圖 1-10　**SYSTEM MASTER** 磁碟片的目錄。標示的圓圈號碼部份會在文章中加以解說。

④ 第 4 欄是檔案名稱。稍後的章節會使用某些檔案。

您將會反覆地利用 **SYSTEM MASTER** 磁碟片上的程式來管理您的各種檔案，稍後，您就學會如何複製本磁碟片。

電腦術語小註：

Catalog ——目錄，記載磁碟片內所含的檔案名稱及其性質。

File ——檔案，儲存於磁碟片上“資訊”之集合體。資訊的內容可能是一般文句、資料或程式。

Lock ——鎖住，鎖住一個檔案的目的是保護該檔案，以預防無意的更改或刪除動作。LOCK 也是一個 DOS 指令。

Unlock ——解鎖，表示解除對某檔案的保護作用。第四章會介紹解鎖和鎖住檔案的指令用法。

複製 SYSTEM MASTER 磁碟片

本部份將指導您如何複製（ Copy ） SYSTEM MASTER 整片磁碟的全部內容，並且如何測試“拷貝版”（ Copy ）的完整性。

複製磁碟片的過程常稱之為“**儲備性拷貝**”（ backing up ）。完成 SYSTEM MASTER 磁碟的“**備份拷貝**”（ backup copy ）後，就可將原版保存在較安全的地方，直接用拷貝版來做其他的事。我們建議您，凡是屬於**系統磁片**（ System Disk ）一類的磁碟均需做好備份拷貝的工作，以預防不測。在毀損拷貝版後仍有份完好的正版。（附帶一提，對於系統磁片最好加貼一紙籤，蓋住磁片邊緣“可以寫入”（ write-enable ）的缺口，形成“寫入保護”狀態。請讀先前的“關於磁碟和磁碟機”一節較詳細的解說。）

要複製一片磁碟，您將使用到 SYSTEM MASTER 內的拷貝程式（ COPYA 或 COPY … ）。其動作原理如下：

拷貝程式（ copy program ）自動地設定空白磁片的格式以接收資訊內容。它在能力範圍內讀取原始磁碟片（ original disk ）

的內容，暫時存入主記憶體內，而後再將該份資料寫入（複製）至空白的磁碟片（或已存檔案的另一片磁碟，但會完全破壞該磁碟的原有內容）。簡言之，讀取原始磁碟內容，暫存至記憶體，再複製至另一片磁碟。反覆數次的讀寫動作後（約六次），就可完成一整片磁碟的拷貝工作。

在本磁碟片內有兩個程式COPY和 COPYA 可以複製整片磁碟，其功能極為相近，祇是適用的語言環境（ environment ）不同，為便於辨認起見，在Applesoft BASIC 情況下請用 COPYA（請留意字尾的A字），若是 Integer BASIC 則請執行 COPY 程式。

記住，您可從螢幕上顯示的提示記號分辨目前所屬的 BASIC 環境。若是個] 記號，表示可執行 Applesoft BASIC 程式；若是個 > 記號，則是可執行 Integer BASIC 程式。

一旦您完整地實習拷貝程序，將會感覺過程簡易而快速，當然了，第一次做會有點混淆不清，所以請您細心地跟隨下列的操作程序，尤其是在本書告訴您直接按下 RETURN 鍵時，要多注意螢幕的變化情形。

在操作之前，有三個螢幕上常見的術語，其意義如下所示：

- ORIGINAL 原始，也即是母版，本例是 SYSTEM MASTER 磁碟片。
- DUPLICATE 副本，即是接收資訊的空白磁碟片。
- DEFAULT 預設值，表示您若不打入不同的號碼，即採用所顯示的數值為適用的 Slot（擴充插槽）或 Drive（磁碟機）代碼。

（註：若在複製過程中出現不尋常的訊息，請查閱“若您有了拷貝問題…”一節說明。）

下列的文章分別討論“僅一部磁碟機”和“二部磁碟機”的情況，請擇其中之一閱讀之。

一部磁碟機的拷貝過程

操作程序	結果
1. 將 **SYSTEM MASTER** 磁碟片插入第一部磁碟機。查驗磁碟機的門是否完全關好。	
2. 查看螢幕上的游標及提示字元，以明白目前所處的語言環境。 倘若是] 提示符號，請打入 [R][U][N][SPACE][C][O][P][Y][A] 倘若是＞提示符號，請打入 [R][U][N][SPACE][C][O][P][Y]	
3. 檢查打字，若無誤則按下 [RETURN] 鍵	按下 [RETURN] 鍵立即執行該拷貝程式。在螢幕上會顯示 APPLE DISK DUPLICATION PROGRAM
4. 連按三次 [RETURN] 鍵，表示您同意接受電腦已預擬的數值。在電腦界常習稱之為“**預設値**”（default）	螢幕上應顯示如下的一段文句： ORIGINAL SLOT: 6 DRIVE: 1 DUPLICATE SLOT: 6 DRIVE: DEFAULT

(**註**："**擴充插槽**"是在 Apple II 主機板內的連接子（connector），用來裝設如磁碟機控制界面卡、印表機之類的周邊裝置界面卡。
一張磁碟機界面卡可以接上一部或二部磁碟機，其作用是控制磁碟機的運轉及讀寫動作。添設"周邊裝置界面卡（peripheral card）"於擴充插槽上是爲了增強或修改電腦主機的能力。）

電腦主機板內有數個**擴充插槽**（expansion slot）拷貝程式是假設**磁碟機控制界面卡**（disk controller card）是插在slot 6，同時將它控制的第一部磁碟機當作原始磁碟機。連按三次 RETURN 鍵，表示您同意（接受）前三項預設值。

5. 按下 1 鍵。告知拷貝程式您的原始磁碟的副本（空白磁碟片）將放在第一部磁碟機。

請注意螢幕上的字樣 DRIVE: DEFAULT = 2已經改變成 DRIVE: 1.。

預設值是2，因爲很有可能兩部磁碟機是同時接在同一插槽上的界面卡上。若您有兩部磁碟機，請讀下一節說明。

現在，拷貝程式顯示目前您能使用的插槽編號及磁碟機代號。
當螢幕出現了 PRESS 'RETURN' KEY TO BEGIN COPY 訊息，表示電腦已明瞭拷貝的來源處和目的處，但仍未正式進行複製的工作。
整個過程的完整畫面應如圖1-11所示，假若您僅有一部磁碟機，但不是如圖 1-11 所示的畫面，則將無法正確地完成複製工作。
請按下 CONTROL - RESET，當 **BASIC** 的提示符號再現於螢幕上，重新第 2 步驟。

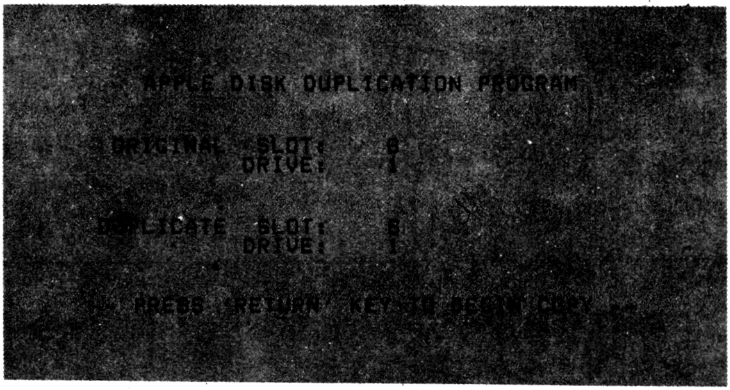


圖 1-11　對應一部磁碟機的拷貝程式畫面

6.	按下 RETURN 啓動拷貝的過程。在進行中，若有不尋常的訊息，請查閱"拷貝磁片出了問題時…"一節。	按下 RETURN 後立即進行複製工作。您將先看到 INSERT ORIGINAL DISK AND PRESS RETURN 訊息，通知您程式已經準備妥當，請把想複製的磁片（正版）放在磁碟機內。若您想複製**SYSTEM MASTER**以外的磁碟片，請在這個時候抽換 **SYSTEM MASTER**，放入另一片想複製的磁碟。
7.	再度查驗 **SYSTEM MASTER**是否在第一部磁碟機。按下 RETURN 鍵告知電腦，您已放入了原始磁碟片。	第一部磁碟機的指示燈亮了，同時在 ORIGINAL SLOT: 6. 旁顯示白底黑字的 READING 。其用意是告訴您拷貝程式正在讀取**SYSTEM MASTER** 內的資訊，並將它暫時存放在主機的記憶體上。 而後，螢幕顯示另一行訊息 INSERT

步驟	動作	說明
		DUPLICATE DISK ，通知您 AND PRESS RETURN，通知您拷貝程式已準備好了，要將暫時存放於記憶體的資訊轉移到另一片空白磁碟。
8.	小心地移走第一部磁碟機內的 **SYSTEM MASTER** 。放入另一片空白磁碟，並按下 RETURN 鍵。	螢幕上 DUPLICATE SLOT: 6 之旁多了白底黑字的 INITIALIZING 字樣，表示電腦爲了準備接收資訊，在空白磁碟上預作安排格式。（稍後，會詳說細節） 接著，在 DUPLICATE SLOT: 6. 之旁顯示白底黑字的 WRITING ，表示拷貝程式正在將記憶體內暫存的資訊轉移至空白磁碟片。 最後，又重現 INSERT ORIGINAL DISK AND PRESS RETURN.訊息。
9.	在第一部磁碟機上抽換原始的 **SYSTEM MASTER** 磁片。 現在，**SYSTEM MASTER** 磁片又在第一部磁碟機上。 按下 RETURN 鍵。	您再見到ORIGINAL SLOT: 6 旁的白底黑字 READING ，同時

	第一部磁碟機的指示燈又亮了，表示程式進行讀出磁片內容，並暫存至記憶體的工作。
10. 等候 INSERT DUPLICATE DISK AND PRESS RETURN. 訊息的重現，移走**SYSTEM MASTER** 磁碟片，並再將副本磁碟片（先前的空白磁碟）插入第一部磁碟機。按下 RETURN 鍵。	您再見到 DUPLICATE SLOT: 6. 旁會出現白底黑字的 WRITING 字樣，表示暫存於記憶體的資訊正寫入磁碟片中。
11. 重複 9. 10. 兩個步驟，一直到螢幕顯示出 DO YOU WISH TO MAKE ANOTHER COPY?	本項訊息告知使用者，拷貝程式已完成複製磁片的任務。現在，您擁有兩份**SYSTEM MASTER** 磁碟片。過程簡易而刺激，對吧?!
12. 在標籤上寫些短句，貼到新拷貝的副本磁碟片。（可參考先前的“照顧好您的磁碟片”一節說明）	

註：要有耐心！在同一部磁碟機複製磁碟須抽換六次磁碟片，才能完成一份 **SYSTEM MASTER** 副本。至於其他種類的磁片，在複製過程中眞正抽換的次數，完全是看原始磁碟片所含的內容多少而定。

13. 假若您想再拷貝一份，請按下 [Y] 鍵，並遵循螢幕上的指示行事。	您不必再重鍵入有關Slot 和 Drive的數值。電腦仍維持在第 6 步驟所選擇的數值。
假若您想複製 **SYSTEM PROGRAMS** 磁片，請在缺口加貼保護標籤（預防抽換磁片出錯），然後在 `INSERT ORIGINAL DISK AND PRESS RETURN` 訊息出現在螢幕上時，將此磁碟片放入磁碟機內。	
若不必再複製多份副本，直接按下 [N] 鍵，表示“NO”，並加按 [RETURN] 鍵，可以繼續閱讀“測試拷貝版磁碟片“一節內容。	在螢幕上會重現您熟識的**BASIC**提示符號（] 或 > 記號）。

註：在拷貝一片有“可寫缺口”（Write-enable notch）磁碟之前，先用不透光貼紙蓋住此缺口，是個很好的習慣。因爲它可保證在忙中有錯的意外情況下，不會危害到原始磁碟。記住，只要“可寫缺口”覆蓋了貼紙，就如同加上“寫入保護”措施（write-protect），只能讀取此磁碟的內容但無法寫入任何資料。

兩部磁碟機的拷貝過程

操作程序	**結果**
1. 將 SYSTEM MASTER 磁	

	碟插入第一部磁碟機。查看磁碟機的門是否已完全關好。	
2.	看看游標，判斷目前是使用那一種 **BASIC** 語言。若是] 提示記號，打入 [R][U][N][SPACE][C][O][P][Y][A] 若是 > 提示記號，打入 [R][U][N][SPACE][C][O][P][Y]	
3.	在檢查打入的指令無誤之後，按下 [RETURN]	按下 [RETURN] 後即啓動拷貝程式的執行，螢幕上顯示 APPLE DISK DUPLICATION PROGRAM 訊息。
4.	連按 [RETURN] 四次，表示您同意接受電腦已預擬的數值。在電腦界常習稱之爲“ **預設值** ”(default)。 電腦主機板子有數個擴充插槽(expansion slot)。本拷貝程式是假設您的磁碟機控制界面卡插在 Slot 6，界面卡連接兩部磁碟機，以 drive 1 爲原始磁碟所在的磁碟機，而 drive 2 對應於副本磁碟。您按下 [RETURN] 四次，即是表	現在，拷貝程式顯示欲使用的擴充槽編號及磁碟機代號。當 PRESS 'RETURN' KEY TO BEGIN COPY 的訊息出現於螢幕時，表示電腦已明瞭拷貝的來源處和目的處，但仍未正式進行複製的工作。 螢幕應顯示如圖 1-12 所示的畫面，除非您眞的有兩部磁碟機，否則無法正確地完成拷貝工作。若您按下 [CONTROL]-[RESET]，則重現 **BASIC** 提示符號，再重新進行第 2 步驟。

示同意這些預設值。

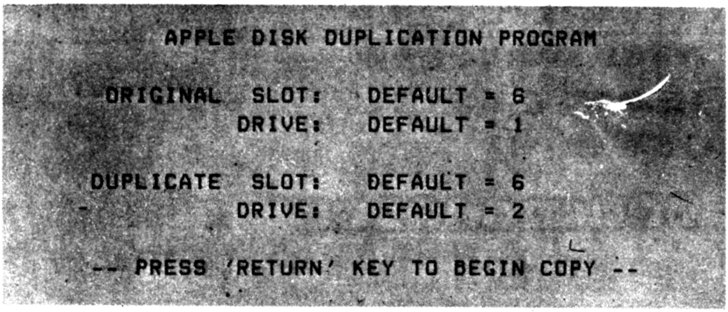


圖 1-12 兩部磁碟機情況下的擴充槽及磁碟機代號

---

5. 在按下 RETURN 鍵之前，各磁碟片務必擺放在正確的磁碟機：原始磁碟在第一部，空白磁碟在第二部。

在程式讀取 **SYSTEM MASTER** 磁片內容並暫存至記憶體時，螢幕上會顯示 READING 字樣。而後的INITIALIZING字樣表示它正在空白磁碟片上預作接受資訊的安排，隨後的WRITING字義告訴使用者，正在複製這一份資訊內容於副本磁碟片。而後，READING 重現，表示讀取 **SYSTEM MASTER** 內另一部份的資訊並暫存於記憶體內。讀取及寫入的動作會重複多次，一直到複製的工作全部完成，在螢幕上會看到親切的詢問句子：DO YOU WISH TO MAKE ANOTHER COPY?（若在複製過程中，有不尋常的訊息，請參閱“拷貝磁片出了問題時…”一

		節說明。）
6.	適當地註明文字於標籤上，貼在新拷貝版磁碟上。	
7.	假若您想再拷貝一份，請按下 Y 鍵。記著要放入另一片空白磁碟於第二部磁碟機，而後遵循螢幕上的指示行事。 假若您想複製 SAMPLE PROGRAMS 磁碟，請先在“可寫缺口”（write-enable notch）加上一銀質的寫入保護貼紙，而後插入 SAMPLE PROGRAMS 磁碟於第一部磁碟機。	您不需要再一次地打入擴充槽及磁碟機的代號。在往後的處理過程會一直沿用第 5 步驟的預設值。
	假若不必再複製多份副本，直接按下 N 及 RETURN 鍵。請接著閱讀“測試拷貝版磁碟”章節的內容。	在螢幕上重現BASIC 提示符號（] 或＞記號）。

註：在拷貝一片有“可寫缺口”（Write-enable notch）磁碟之前，先用不透光貼紙蓋住此缺口，是個很好的習慣。因爲它可保證在忙中有錯的意外情況下，不會危害到原始磁碟。記住，只要“可寫缺口”覆蓋了貼紙，就如同加上“寫入保護”措施（Write-protect），只能讀取此磁碟的內容但無法寫入任何資料。

拷貝磁碟出了問題時

**** UNABLE TO READ ***** 或I/O ERROR等錯誤訊息出現時，其原因可能是 SYSTEM MASTER 磁碟沒放入第一部磁碟機，或該部磁碟機的門沒完全關上，或者是插槽代號及磁碟機編號弄錯了。

倘若是***** UNABLE TO WRITE ***** 或 I/O ERROR 等訊息，可能是您沒有將副本磁碟片正確地擺放在磁碟機內，或者該磁碟片已加上了寫入保護的標籤（請參考"關於磁碟片和磁碟機"一節說明），或該部磁碟機的門沒完全關好，或是插槽代號及磁碟機編號弄錯了。

假若 COPYA 程式無法讀取原始磁碟的內容，或無法寫入資訊至副本磁碟片，則您會被詢問" DO YOU WISH TO MAKE ANOTHER COPY? "，請斷然拒絕它，按下 N 鍵，會重現 Applesoft BASIC 提示符號]。

假若COPY程式無法"啓始化"(initialize)副本磁碟片，先出現的是 I/O ERROR 錯誤訊息，接著是個 Integer BASIC 的提示符號>。若COPY無法順利地完成磁片間的資訊轉移任務，會顯示出 DO YOU WISH TO MAKE ANOTHER COPY?，請打下 N 鍵及 RETURN 鍵。螢幕上重現 Integer BASIC 提示符號。

總而言之，在重新嘗試複製磁碟之前，先逐項檢查下列各事項:

- 您的 SYSTEM MASTER 磁碟妥當地放在第一部磁碟機(drive 1)內。
- 磁碟機的門完全關好。
- 拷貝版的空白磁碟有個"可寫缺口"（它不應覆蓋"寫入

保護”貼紙）。

- 使用正確的擴充槽編號及磁碟機代號。
- 若有一部磁碟機，則遵循“一部磁碟機的拷貝過程”的操作指示；若有兩部磁碟機，則遵循“兩部磁碟機的拷貝過程”。

在檢查上述事項之後，請回至第2步驟重新操作，再試一次。假若仍然有問題，也許是空白磁碟片已毀損，換其他的磁碟片試試看。若一直無法順利拷貝磁碟，請與您的電腦經銷商聯絡。

測試拷貝版磁碟

當您已有片複製過的副本磁碟，我們建議您查證一下。倘若它不是**系統磁碟**（system disk，意在開啓電源時，可自動啓動主機的磁碟），您祇要查驗它的目錄，比對一下是否即是原始磁碟內的檔案名稱，若完全一致，即是成功的磁碟片。若顯示了錯誤訊息或是目錄部份不完整，請再嘗試拷貝程序一次。提醒您，打入

C A T A L O G

並按下 RETURN 鍵可獲知磁碟片內所含的檔案名稱（目錄）。

假若新磁碟是屬於系統磁碟，您必須重開電源（或 CONTROL – RESET 鍵）來重啓電腦，來測試這片拷貝版磁碟。例如，要測試您複製的 **SYSTEM MASTER** 副本是否完好，請將此磁碟片放入第一部磁碟機內，看看它能否啓動**DOS**。

本章節將討論兩種重啓電腦的方法，不過它與您的 Apple II 機型有關。其他方法則在第5章介紹。

- 若主機是 Apple IIe 型，兩種程序均適用。
- 若主機是 Apple II 或 Apple II Plus，僅第二種程序適

用。

第一種：

下列的方法可以不必讓您反覆地關閉又重啓 Apple IIe 主機，節省您的時間，同時也損少電腦機械和電子零件的損耗。

操作程序	結　　果
1. 將新拷貝版的 SYSTEM MASTER 放在第一部磁碟機，並關好磁碟機的門。	
2. 同時按住 [Apple] 和 [CONTROL]，並按下 [RESET]。	第一部磁碟機開始運轉（燈會亮著），稍候，燈滅了，也沒有運轉聲音。
放掉 [RESET]，而後是 [CONTROL]，最後鬆手 [Apple] 鍵。	若 SYSTEM MASTER 磁片已正確地被複製成功，螢幕上會出現如前所述的圖 1-9 畫面。若不是，請回至"複製 SYSTEM MASTER 磁碟"一節再重新拷貝一版。

（本法僅適用於主機的鍵盤上標示有 [Apple] 鍵的 Apple IIe 機型。）

第二種：

下列的方法可以重新啓動 Apple II 的三種機型（Apple II，Apple II Plus 和 Apple IIe）。只有能成功地重啓在新拷貝版的程式，才能證實它是好的磁碟。

操作程序	結　　果
1. 伸手到主機左半機座，找到電源開關，並關掉它。（若不知開關位置，請參考圖	

	1-8。)	
2.	將新拷貝版 SYSTEM MASTER 放在第一部磁碟機，並關好門。	
3.	開啓主機的開關。	第一部磁碟機指示燈亮起，並開始運轉。稍候會兒，不再運轉，燈隨即熄滅。 若 SYSTEM MASTER 磁片已正確地複製成功，螢幕上會出現如前所述的圖 1-9 畫面。倘若不是，請回至"複製 SYSTEM MASTER 磁碟"一節再重新拷貝一版。提醒您，若在開啓開關後，螢幕出現的是星號（ * ），請打入 [6] [CONTROL]-[P] 以重啓 DOS 。

（註 1 ；若在螢幕上見到星號，請查閱先前"解決啓動的問題"內"沒有任何顯示訊息或有奇怪訊息"章節說明。

註 2 ：若想多了解啓動的方法，可查閱第 5 章的" PR ＃指令"或"啓動 DOS 的其他方式"章節解說。）

預備一空白磁碟片－INIT指令

到目前爲止，您已學會如何啓動 DOS ，如何複製一磁碟內所有東西到一新磁碟片。現在，您該準備好一空白磁片以便能接受（儲存）資訊。

當您從經銷商處購得空白磁片，它們是沒有記錄任何東西在內的，就像您買到的空白錄音帶一樣的情況。在一空白磁碟片能接受資訊之前，您必須用 INIT 指令作好儲存的事前準備工夫。這一道預作準備的手續，我們常稱之為〝**啓始化**〞（ initializing ）或“〝**規劃格式**〞（ formatting ）。

這一道手續需要兩個步驟，首先是寫一段“歡迎詞”程式（ greeting program ），而後才下達 **INIT** 指令。

取名“歡迎詞”的意思，是因該程式在啓動磁碟後，程式自動被執行，在您的感覺上是“它”首先出面跟您說聲哈囉！　。大家常為此類的歡迎詞程式，命名為 **HELLO**。會如此命名的來源是建立在共同的默契，在啓始化一空白磁碟時，儘量用簡易又有意義的名稱，易記好用。

下文會告訴您如何建立一個為啓始化——空白磁碟的“歡迎詞”程式，至於這程式的內容依各人喜好而定。通常“ hello ”的程式僅是一單列的（ BASIC ）程式，而如 **DOS 3.3** 的 **SYSTEM MASTER** 磁碟片，其情況較複雜些。

警告：當您想啓始化一磁碟片，即是重新規劃磁碟的格式，在磁碟內的所有資訊（資料或程式）將會被消除掉。

請依循下列的磁碟啓始化步驟。

操作程序	**結果**
1. 自第一部磁碟機取走 **SYSTEM MASTER** 磁片，放入一片空白磁碟並關好門。注意，該磁片不許在“可寫缺口”（ Write-enable notch ）加貼標籤。	

2. 打入

NEW

並按 RETURN 鍵，先清除在記憶體內所存的任何東西。

3. 現在，請您打入如下的小小程式。在每一列之後已確定打字無誤後，才按下 RETURN 鍵。請用您的名字和今日日期取代列號 20 內的名字、日期部份。請小心地打入空白和雙引號。若弄錯了，在按下 RETURN 前可用 ← 鍵倒退游標去更正錯誤。

螢幕上的程式應如下所示：

```
10 PRINT "THIS HELLO PROGRAM CREATED BY"
20 PRINT "(your name) ON (date)"
30 END
```

本程式將由 INIT 指令喚出以執行規劃出一可用的空白磁碟片的工作。而後，每一次磁碟片啓動時，它會自動執行之。

4. 測試您剛打入的程式，請打入

RUN

螢幕上即顯示您打入的大名和日期。

	並按下 RETURN 鍵。	
5.	測試無誤後，請打入 I N I T SPACE H E L L O 並按下 RETURN 鍵。 HELLO 即是上述的三列程式的名稱。 INIT 指令是 DOS 其中一個指令，它後隨一個程式名稱，其目的是告訴作業系統，在啓始化一空白磁碟時，將該程式的內容（本例即是 HELLO 的三列 BASIC 程式）存入磁碟內，並視之爲啓動磁碟後的歡迎詞程式。	直等到磁碟機已停止嘈雜的聲音，指示燈滅了，同時提示符號再現於螢幕上，表示已結束了磁碟片啓始化的工作。 若您得到的是 I/O ERROR 錯誤訊息，檢查您的磁碟片（不應貼住可寫缺口）或磁碟機（門是否完全關好，或磁碟是否放反了。），再重試 INIT 指令。

6.	測試本磁碟以查證它確實已規劃格式了。打入
	C A T A L O G
	並按下 RETURN 鍵。
		螢幕上應有如下的簡短目錄：
		`DISK VOLUME 254`
		`A 002 HELLO`
		DOS 系統已指派您的磁片其卷號是 254（它是標準的預設值，除非您特別規定不同的數目），並將您的 BASIC 程式寫入其內，在目錄中您可獲知該程式是名爲 HELLO，其長度爲 2（002）磁區。

7.	細心地移走磁碟機內的磁碟並附貼一張標記貼紙。請參	

考圖1-1所示的標記位置。
您可以在標記紙上作個記號
，表示它已是啓始化過的。
當您不想用它時，放入**磁碟封套**（ disk envelop ）以
避免灰塵的污損。

現在，您預備好的磁碟片已通過測試，表示它隨時可與 DOS 系統配合使用，以儲存程式、資料、文句、圖形或者其他您希望的資訊內涵。暫時讓可寫缺口保持原狀，好讓您隨時寫入資訊並保存於磁碟上。

（**註**：任何啓始化的磁片的性質是與主機的記憶體大小有關。系統當初規劃磁片格式時的記憶體大小，即決定了再使用此磁片的主機內最低限度的記憶體。例如，若一磁碟是在 32 K的主機被規劃格式，它便只能在有 32 K以上記憶體的主機使用。）

電腦術語小註：

initializing ——啓始化，有時也稱之規劃格式（ formatting ），其目的是在爲一空白磁碟作些準備工夫，方能接受資訊，儲存在磁碟之內。

INIT Command —— INIT 指令，其目的是啓始化一磁片，並儲存 DOS 系統的程式（以便可以啓動磁碟）及歡迎詞程式在磁碟內。

NEW Command ——NEW 指令，其目的是爲一新程式預作準備，清除記憶體內既存的所有東西。

PRINT Command —— PRINT 指令，是屬於BASIC 的指令，執行BASIC 程式，一遇到 PRINT 敍述，可顯示首尾雙引號之間的文句。

END Command —— END 指令，亦是 BASIC 指令，用來表示程式的終止點。

使用程式

某些程式（例如電動遊戲程式）在您啓動磁碟機時，會自動地執行，根本不需要您的介入就會顯示些訊息，接著遵循您從鍵盤打入的指令，從磁碟內取出某種資訊，或將資訊儲存於磁碟片上。然而，在某些情況下，您必須知道程式究竟是如何做事的，尤其您有必要了解如何將磁碟內的資訊存入主機的記憶體內（此步驟常稱之爲〝**載入程式**〞（Loading）），如何修改已存於記憶體內的資訊（此步驟稱之爲〝**編輯**〞 Editing），以及如何將記憶體內的資訊儲存於磁碟片上（此步驟稱之爲“**儲存程式**” Saving）。

假若您想作進一步的程式設計，就得學習如何使用指令來指引電腦從事您希望的工作。下列的章節將告訴您載入，編輯和儲存等相關事項。

自磁碟中讀取資訊—LOAD 指令

要查看磁碟片內某一檔案內含的資訊內容，您首先必須將檔案的內含資訊複製一份至主機的記憶體內。這一個處理程序常稱之爲“**載入**”，而以LOAD指令來完成這個步驟。

操作程序	結　果
1. 將 SAMPLE PROGRAMS 磁碟片放入第一部磁碟機。	
2. 打入	磁碟機開始運轉，找到並讀取

LOAD SPACE VERIFY.ME	VERIFY.ME 檔案內含的資訊，將該份資訊存入記憶體內。
並按下 RETURN 鍵。	若程式已載入記憶體內，則 BASIC 提示符號會再現於螢幕上。 倘若顯示的是 FILE NOT FOUND（未尋著該檔案），請重新鍵入正確的指令。請注意本例中的句點，也是檔案名稱的一部份。
3. 查看程式的內容，打入 LIST 並按下 RETURN 鍵。	在螢幕上會看到VERIFY.ME檔案的內容。實際上，它僅是一列的 BASIC 程式： 10 REM VERIFY.ME

（**記憶體內涵不是永恒不變的**——一旦您關掉電腦的電源時，自磁碟片讀取並存入記憶體的資訊立即消失不見。然而，您祇不過是將（檔案的）“副本”存入記憶體內，即使它消失不見了，任何時候您仍有份磁碟片上的“正本”檔案。）

改變記憶體的內含資訊

要修改既存檔案的內容，您首先必須先載入一份該檔案內容的副本至記憶體。現在，既然已經將 VERIFY.ME 的程式載入記憶體，讓我們學會如何改變記憶體的內含資訊，並如何儲存“新版本”程式於磁碟片上。

（**檔案與記憶體**：所謂的“檔案”（file），是一些資訊的集合體，給予一名稱，放入電腦的週邊儲存裝置，此裝置常是個磁碟機。換句話說，一提到“檔案”，即是意味它是“磁碟”上的某種東西。若將相同的資訊內容儲存於記憶體內，我們不會稱呼它是“檔案”，而是依據該資訊的性質，稱之為文件（document），程式（program），圖畫（picture），運算圖表（worksheet）或資料（data），或者其他適當的稱謂。）

操作程序	**結果**
1. 移走 SAMPLE PROGRAMS 磁碟片，並將先前已規劃過格式的磁碟片放入第一部磁碟機。	
2. 確定是否放對了磁碟片，打入 [C][A][T][A][L][O][G] 並按下[RETURN]鍵。	此新磁碟片的目錄，指明磁碟片內僅含一程式 HELLO。
3. 打入 [L][I][S][T] 並按下[RETURN]鍵。	您發現 VERIFY.ME 程式仍在記憶體內。除非您曾經關掉電腦，應再將它載入記憶體。 它是否仍是如下的一列程式？ `10 REM VER. Y.ME`
4. 現在，再多增加一列BASIC敍述，如下所述： `20 PRINT "NEW LINE"`	在按下 [RETURN] 鍵後，會將列號 20 加入已存在於記憶體的 VERIFY.ME 程式。然而，此舉並不會改變在磁碟片的該程

記得要按下 RETURN 式。
鍵。

---

5. 再增加一列　　　本列會加入記憶體內的程式。

```
30 END
```

並按下 RETURN 鍵。

6. 查驗一下這個新程式，請用 LIST 指令並按下 RETURN 鍵。　　　記憶體內的程式，如下所示：

```
10 REM VERIFY.ME
20 PRINT "NEW LINE"
30 END
```

現在，您已經新增了兩列在既存於記憶體的程式。只要讓電腦的電源一直開著，也不去下達 NEW 指令，同時也不試圖以其他 BASIC 指述取代它們，那麼 VERIFY. ME 程式會一直存在記憶體內。假若您想一而再地使用這程式，就必須將這一份存入磁碟片。下一章節將告訴您如何儲存程式。

電腦術語小註：

Loading ——載入，將磁碟片的內含資訊，複製一份並轉移至記憶體的過程，稱之爲" 載入程式 "。

Editing ——編輯，改變或修改記憶體的內含資訊之過程或動作，稱之爲" 編輯 "。

Saving ——儲存，將記憶體的目前內含資訊，複製一份轉移至磁碟片的過程，常稱爲" 儲存程式 "。

LOAD Command ——LOAD指令， DOS 的指令，用以完成" 載入程式 "動作的指令。

儲存資訊至磁碟片— SAVE 指令

將一程式存入磁碟內，當作“檔案”永久地保存下來，或是一直保存到您想改變這程式的時候。SAVE 指令的目的是儲存任何已有過修改的程式（例如，在歡迎詞程式裏修改日期），或已增添過內容的程式（例如，在電話名冊程式內加入新的號碼），或是您剛剛設計完成的新程式。一旦您有了“儲存”這個念頭，您必須準備好一片可寫入資訊的磁碟片，也就是該磁碟片不得爲“寫入保護”。請用已規劃過格式的磁碟片進行下列的步驟。

操作程序	結　果
1. 將記憶體內程式存入磁碟片，請打入 S A V E SPACE P R O O F SPACE P O S I T I V E	若得到的是 SYNTAX ERROR 錯誤訊息，請再試一次，也許是您打入的 SAVE 字語不完全是大寫字母。
並按下 RETURN 鍵。	當 DOS 將記憶體內含資訊複製一份，以 PROOF POSITIVE 檔名儲存於磁碟片時，磁碟機保持運轉並且指示燈閃滅數回。現在，修訂過的程式（列號 10 至 30 ）仍在記憶體內，同時也存一份在磁碟上。

注意： 若一檔名想含有兩個字語（ word ）時，如本例的 PROOF 和 POSITIVE ，請在兩字語之間打入“空白字元”。記住，一“空白”對於電腦而言，它的地位是視同英文字母或數字的

“字元”（character）。若您需要的是個“空白字元”，請按下 SPACE 空間棒（空白鍵）。

2.	查驗是否程式已存入磁碟片，請下達 **CATALOG** 指令並按下 RETURN 鍵。	螢幕上顯示如下的訊息： DISK VOLUME 254 A 002 HELLO A 002 PROOF POSITIVE 沒錯！**PROOF POSITIVE** 程式確實已經成功地自記憶體內轉存至磁碟片，成為一“檔案”。
3.	打入 N E W 並按下 RETURN 鍵。	主記憶體已清除程式。若您下達 **LIST** 指令，只會回應一個 **BASIC** 提示符號。現在，您可以開始撰寫（設計）新程式或載入磁碟內任一檔案。
4.	載入程式，以確認剛剛已完成了儲存動作。打入 L O A D SPACE P R O O F SPACE P O S I T I V E 並按下 RETURN 鍵。	**DOS** 會將磁碟內的該程式，複製一份至記憶體內。
5.	打入 L I S T 並按下 RETURN 鍵。	**PROOF POSITIVE** 程式的內容全顯示於螢幕。

附註：最原始的一列程式 VERIFY. ME 仍舊保存在SAMPLE PROGRAMS 磁碟片上。而您是用另一個名稱PROOF POSIT-

IVE 儲存三列程式於磁碟片上。所以，您是同時保有兩個版本的程式。

任何時候，您修改了一個程式，假若想保存兩個版本的程式，您必須用不同於原先檔名的新名稱加以儲存之。倘若仍沿用原先的檔名，則新版本的程式將會“覆蓋”在舊版本上；換句話說，舊版本程式已被消除不見。

執行磁碟內的程式— RUN指令

在啓動磁碟時，有許多程式是無法自動啓動並執行的。這一類的程式必須藉助於 DOS 的 RUN 指令才能誘導它作事的。

操作程序	結　果
1. 將磁碟機內磁片移走，並插入 SAMPLE PROGRAMS磁片至第一部磁碟機。	
2. 打入 R U N SPACE C O L O R SPACE T E S T 並按下 RETURN 鍵。	DOS 立即進行搜尋，載入並啓動 COLOR TEST 程式。這時候，您可以看到有一長列的 COLOR TEST 的功能表選擇項（ menu ）供您挑選。
3. COLOR TEST 程式的目的是讓您測試“彩色”螢幕的特色。假若您沒有彩色的顯示裝置，您看到的僅是黑白變色的螢幕畫面。本程式列舉操作概要於螢幕上，您可以隨意選擇之，打入	

	1, 2, 3, 或 4 並按下 RETURN 鍵。	程式開始"展開"它的功能。
4.	按下 RETURN 鍵可結束任何一項的展示活動並回至原先的功能表。	螢幕重現功能表。
5.	結束 COLOR TEST 程式的執行，打入 5	BASIC 提示符號再度重現於螢幕上。

RUN 指令是個使用頻繁的 DOS 指令，可以讓您啓動一程式的執行動作。若您想測試它的作用，您不妨試試 SAMPLE PROGRAMS 內的其他程式。某些 Apple II 的老舊機型在執行COLOR TEST 程式時，也許會出現 LANGUAGE NOT AVAILABLE 訊息，假若眞有此情況發生，請改試APPLEVISION 程式，並遵循螢幕上的指示操作之。

關掉電腦

現在，您已完成短短的自習課程，可以放下電腦休息一下了。請依循下列的步驟關掉電腦：

1. 取出磁碟片並關好磁碟機的門，請檢查是否磁碟機內的磁碟片已取出。
2. 伸手至電腦機箱後部，關掉電腦的電源開關。
3. 關掉您的顯示裝置（顯示器或電視機）。

到此爲止，您完成本自習課程，若自覺習得的指令其操作法不

夠熟練，不妨再回頭重讀本章並加以演練之。若興緻正高，那麼可以執行 SAMPLE PROGRAMS 內的其他展示用程式（demonstration program）滿足您的好奇心。本書的附錄C會解釋磁碟內各種程式的意義或目的。請自行挑選您喜愛的程式加以測試之。

若您準備好再深入探討 DOS ，那麼第二章可以概略性地介紹 DOS 的全貌，讓您了解啓動 DOS 時，在那畫面之背後有那些事項值得學習，DOS 與其他程式的關係。

電腦術語小註：

RUN Command —— RUN 指令，它會找尋在磁片內的某一指名的BASIC 程式，將其內容複製一份載入記憶體內，而後隨即執行該程式。

Menu ——功能表，顯示於螢幕上的畫面，在其上列舉了一長串的選擇項，供使用者挑選其中的某項功能。

第1章摘要

本章的目的是指導您使用 DOS 的基本操作法。除了再增添一些專用術語外，您同時也學會下列的事項：

- 保管（照顧）磁碟片。
- 以 DOS 啓動電腦。
- 自磁碟機放入或取出磁碟片。
- 利用CATALOG指令查看磁碟片的內容（目錄）。
- 利用 SYSTEM MASTER 磁碟片內的“拷貝程式”複製磁碟片。
- 測試一拷貝版的磁碟片：查看該磁碟片的目錄部份，或者

測試它是否能重新啓動電腦。

- 為一空白磁碟片創製“歡迎詞程式”。
- 利用 INIT 指令準備好一可接收資訊的空白磁碟片。
- 利用LOAD指令，將磁碟片某檔案的內容複製一份至記憶體內。
- 利用SAVE 指令，將記憶體內的資訊儲存至磁碟片。
- 利用 RUN 指令，啓動一已存於磁碟片上程式的執行動作。
- 如何關掉電腦。

以乎已學得不少東西！想一想，用什麼方式獎勵自己的努力成果？

2

DOS概觀

■DOS如何動作　61
●監督程式　62
●啓動的過程　62
△啓動的磁碟機　65
△啓動用磁碟片　65
△兩套BASIC語言　66
■檔案及磁碟片卷號　68
●檔案　68
△檔案名稱　69
●卷號　70
■DOS指令　71
●語法　71
△選擇項　73
△預設值　73
△數目　74
●引數　75
△檔案名稱：fn　75
△擴充槽編號：[, Sn]　76
△磁碟機代號：[, Dn]　76
△卷號：[, Vn]　77
△不正確的引數造成的錯誤訊息　78
■第2章摘要　78

第2章

DOS概觀

DOS 是什麼？它是一種程式，各指令的組合體，其目的是掌握磁碟片各檔案的動向（存取動作），處理磁碟片和記憶體之間的資訊流通情形，並兼作其他的磁碟管理工作。DOS 程式是存在 SYSTEM MASTER 磁碟片內。

本章是往後各章的一般性介紹，會告訴您 DOS 如何工作，以及使用 DOS 各指令的方式、語法（syntax），同時也說明“檔案”、“卷號”（Volume）和 DOS 之間的關係。了解在您的 Apple II 電腦如何用 DOS 配合使用其他程式是非常重要的。在啟動 DOS 後，會發生怎樣的情況？如何確認您進行的動作或資訊流程？或者如何知道 DOS 已準備好接受您的指令？在本章中，您會找到上述各問題的答案。

在第一章的自習課程裡，也許您會對 DOS 的工作情況有不少疑問，本章或許可以給您滿意的答案。

DOS如何動作

當您啟動了 DOS，表示您加入了 DOS 指令及相關的計算能力到您的電腦內。一旦您將 DOS 程式“帶入”主機的記憶體時，您

就可以執行存在於磁碟片的程式，也能夠將各種資訊或程式儲存於磁碟片上。

每一次您想啓動系統，並且使用磁碟片內各種資訊時，您就得將 DOS 帶入記憶體。要使用 DOS 指令，您下達的指令必須是DOS能接受的正確語法格式。

監督程式

監督程式（ Monitor Program ）是電腦的內置式（ built - in ）計算及控制能力，它的地位彷若是高高在上的監督者，控制所有其他的程式。較重要的任務是負責啓動既存於ROM內的 BASIC 程式，以及將 DOS 讀出並存入記憶體內。監督程式本身是放在一自己專屬的記憶體部份，與 BASIC 和 DOS 存放的位置分離不共用。

監督程式也管制到應用程式與鍵盤、顯示螢幕之間的資訊傳輸的問題，以及電腦主機與**週邊裝置**（ Peripheral devices ）——磁碟機和印表機配合工作的事情。

啓動的過程

當您開啓Apple電腦的電源開關時，其內的電子和機械部份立即有反應，將您的 DOS 程式帶入**主記憶體**（ main memory ）。

一旦電腦供給了電源後，它立即會查驗是否主機已接上磁碟機。倘若確實接上了磁碟機，電腦會嘗試將“第一部磁碟機”（ drive 1 ）內的磁碟片系統資訊放至主記憶體。這些查驗兼啓動的工作是由監督程式負責的。

假若第一部磁碟機內的磁碟片含有DOS程式，則DOS資訊立即轉移至記憶體內，並由監督程式啓動 DOS 系統。如果第一部磁

碟機的磁碟片是 **DOS 3.3　SYSTEM　MASTER** ，則會出現如圖 2-1 所示的螢幕畫面，同時 **DOS** 會嘗試將 **BASIC** 程式放置在記憶體內，而後 **DOS** 令 **BASIC** 開始運作（您在螢幕上會看到 **BASIC** 的提示符號）。

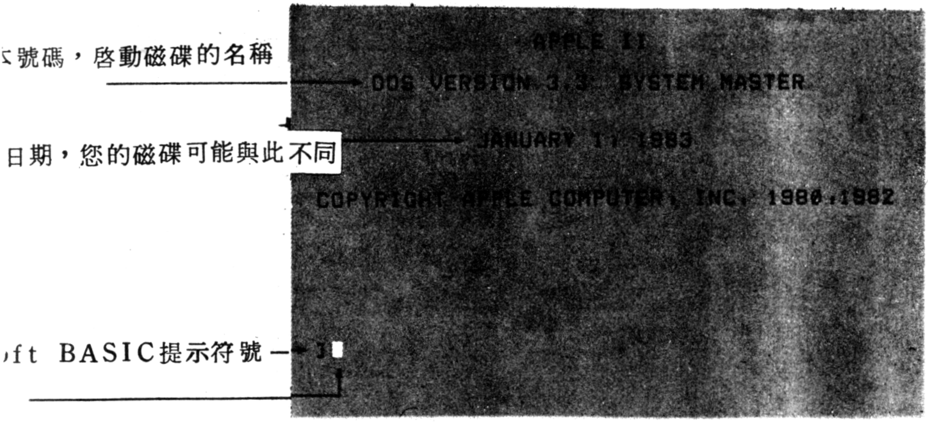


圖 2-1　啓動後的畫面

BASIC 會在螢幕上印出一 **提示字元**（ prompt character ），一般常稱之爲 **提示符號**（ prompt ），以告知您電腦已經準備妥當可以接收您的指令了。

（**註**：若您的Apple主機是舊機型，則配備的可能是無法自動啓動（ autostart ）的舊版監督程式。這一機型的電腦在啓動**DOS**時，在螢幕上僅能看到一個星號（ * ））。
若想進行啓動的過程，您可以打入

6 CONTROL - P

以啓動 **DOS**。）

圖 2-2是啓動的過程中，螢幕的變化情形。

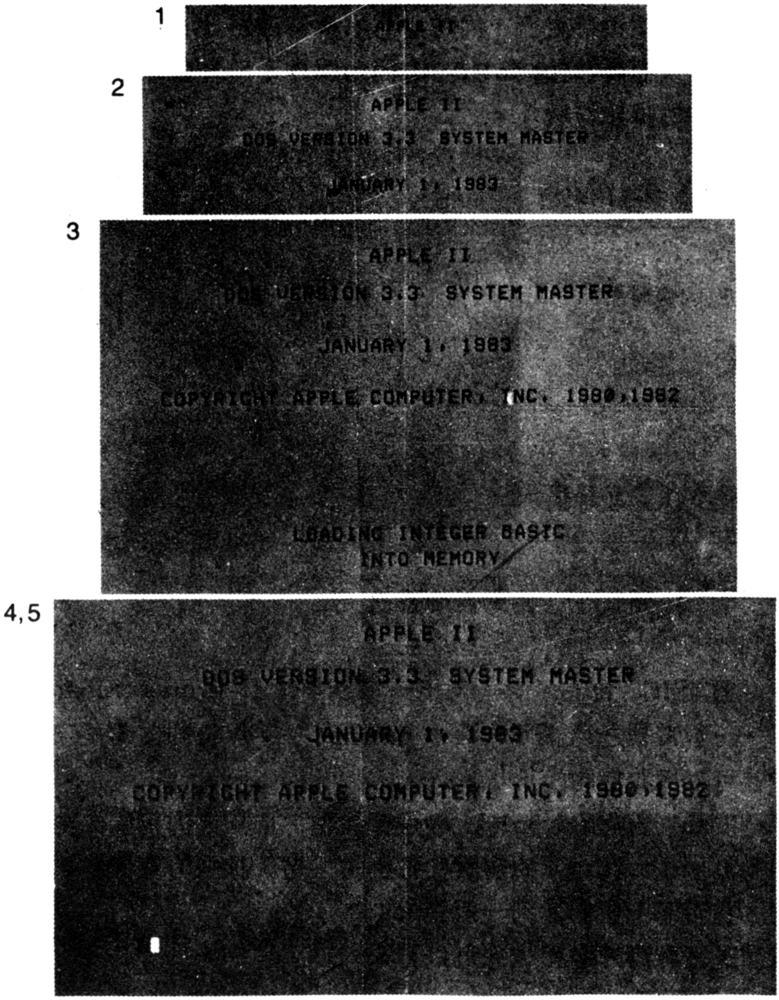

1
2
APPLE II
DOS VERSION 3.3 SYSTEM MASTER
JANUARY 1, 1983
3
APPLE II
DOS VERSION 3.3 SYSTEM MASTER
JANUARY 1, 1983
COPYRIGHT APPLE COMPUTER, INC. 1980,1982
LOADING INTEGER BASIC
INTO MEMORY
4,5
APPLE II
DOS VERSION 3.3 SYSTEM MASTER
JANUARY 1, 1983
COPYRIGHT APPLE COMPUTER, INC. 1980,1982

圖 2-2 啓動的過程

1. 監督程式將 DOS 帶入記憶體並開始執行它。
2. DOS 查驗既存的記憶體容量，並檢查是否另一套BASIC（本例是 Integer BASIC）可供人使用。
3. DOS 執行 HELLO 程式，它是開機即執行的程式。
4. 若記憶體的容量夠大，足以容納另一套 BASIC 程式，即將它載入記憶體內。
5. DOS 啓動 BASIC，而 BASIC 印出一提示符號（] ）。

啓動的磁碟機

在 Apple II 主機內的“自動啓動ROM”會自動嘗試從編號較大的**擴充插槽**（ expansion slot ）找起，看看是否已接上磁碟機，它會先從擴充槽 7（ slot 7 ）進行檢查。若擴充槽 7 是空的，自動啓動 ROM 程式則檢查擴充槽 6；您的磁碟機控制界面卡（ disk controller card ）應插在擴充槽 6。（註：下文中“擴充插槽”均簡稱“擴充槽”）

注意，第一部磁碟機的信號綫（ cable ）應接在磁碟機控制界面卡標誌， DRIVE 1 的連接端子（ connector）上；同理，第二部則接在DRIVE 2 的連接端子上。

倘若如上所述的配備狀況，電源開啓後，第一部磁碟機內的磁碟片資訊即被存入記憶體內。而後，本書均稱“第一部磁碟機”（ drive 1 ）爲“啓動用磁碟機”（ startup drive ）。

假若您不知道那一部磁碟機是“第一部”，又無任何標示可作識別，則請您開啓電腦，看看那一部磁碟機的指示燈最先亮起，便能判斷那一部磁碟機是“啓動用磁碟機”。對於電腦而言，第一部磁碟機即是 “**啓動用磁碟機**” ，常簡稱爲“ drive 1 ”。

啓動用磁碟片

所謂的 “**啓動用磁碟片**” （ startup disk ）是意味著該磁碟

片能在開機後進行讀取資訊時，立即自動執行其中的某一個程式。DOS的指令 INIT 即是用來設定一可啓動的磁碟片。啓動用磁碟片裡面內含著某些必要的資訊，它們可以讓電腦將某一程式自磁碟片複製出來並抄錄在記憶體內，隨即執行該程式。上述章節的 SYSTEM MASTER 和 SAMPLE PROGRAMS 磁碟片均是可啓動的磁碟片，它們會自動的執行 DOS 程式。

附帶一提的，英文字 Starting 與 Booting 在本書是同樣的意義，均譯爲“啓動”。在其他相關的手册或書籍常提到“ boot your system ”或是“ boot a disk ”，其實質意義仍是“啓動一系統”，請讀者不必在這兩個術語打轉，花費不必要的時間。Boot一字起源於Bootstrap ，其意爲不借外力，用自己本身的能力“開始進行”某項事物。

兩套BASIC語言

在您的Apple II電腦能使用那一套 BASIC 語言，是決定於主機的記憶體大小，以及主機的**硬體**（ hardware ）和**韌體**（ firmware ）功能。大多數 Apple II 機型是可以同時具有兩套 BASIC 語言——Integer和 Applesoft 。

當您開啓電腦的電源時，能立即讓您使用的 BASIC 語言，我們稱它爲“**長駐性**”（ resident ）語言。當電腦系統一啓動後，它就自動啓動，進入待命狀態，準備接受您的指令。較早期的 Apple II 電腦是將 Integer BASIC 存放在“唯讀記憶體”ROM（ROM是代表 read-only memory ），而 Apple II Plus 及 Apple IIe 電腦是將Applesoft BASIC 存在ROM內。會稱呼它“長駐性”語言的原因是，該語言是“永久”存放在電腦內（在唯讀記憶體 ROM 中）。（註：若是 Apple II或Apple II Plus機型，您可以選購一“語言界面卡”（ language card ），擴充主機的“隨意存取之記憶體”（ random-access memory ，簡稱

RAM)。語言界面卡並不適用於Apple IIe 機型。)

當您啓動電腦而且主機內有足夠的記憶體(至少 64K)容納兩套語言時，DOS 會從磁碟片內讀取另一套非長駐性 BASIC 語言，將它抄錄至額外的 RAM 部份或是擴充的語言界面卡上。在啓動 DOS後，DOS會回報已抄錄至記憶體的語言類別，例如Applesoft BASIC 若屬於主機的長駐性語言，則在螢幕上會顯示 LOADING INTEGER BASIC INTO MEMORY 反之，若 Integer BASIC 屬於長駐性，則出現的是 LOADING APPLESOFT BASIC INTO MEMORY 訊息。假若您的電腦沒有足夠的儲存位置(記憶體)以容納另一套 BASIC 語言時，則在螢幕上無任何訊息。至於 LANGUAGE NOT AVAILABLE 訊息的意義，請查閱附錄B的說明。

電腦術語小註：

Hard ware ——硬體，意味著一電腦系統內實質的(機械類或電子類)部份。

Firmware ——韌體，意味著一電腦系統內永久存在於唯讀記憶體內的程式部份。

read-only memory ——唯讀記憶體，簡稱 ROM 記憶體的一種，其內所含的資訊僅允許被讀取，但無法再度寫入或加以清除內容，除非您是眞正地自“積體電路”板子(integrated circuit)取走它。

Ranaom-access memory ———隨意存取之記憶體，是可讀取、可寫入的一種記憶體。簡稱 RAM 。在電腦電源關閉或中斷時，存於 RAM 內的資訊會隨即清除不見。

Integrated circuit (或 chip)—— 積體電路，有人簡稱 IC 晶片，是建置於一極微小的矽片內的衆多電路元件之電子產品。

檔案及磁碟片卷號

本部份會說明 **檔案**（ file ）、**卷號**（ volume ）的意義以及兩者相關於磁碟片的事項，同時也提出命名檔案的一般性規則。

檔　案

DOS 將每一片磁碟上的各種資訊加以整理、組織成各個“檔案”。您不妨想像下列的類比例：磁碟片各檔案就有如文件櫃中各個抽屜內的折疊式文書紙夾（ folder ），而文書夾內含有特定的資訊。例如，一標示著“人事資料”的抽屜內，可能有個文書夾專門收存著公司內專職（ full - time ）的員工名册，另一文書夾則記載著給薪職銜的薪資對照表，另一文書夾則是稅率核算表。

若您正想將文件櫃加以電腦化，標示 Personnel（人事資料）的抽屜，就成了 PERSONNEL 卷宗（ volume ），而每一個文書夾就是電腦的檔案，而且給予一適當的專用名稱，例如 EMPLOYEES（表示其性質是員工名册）， SALARY（薪資表）和 TAXES（稅率表）。

任何形式的資訊均可放在檔案內，包括電話號碼，收據、備忘錄、圖畫、程式和資料。

文書紙夾和檔案之間的重大差異是“效率”問題。

- 磁碟檔案可在小空間內儲存大量的資訊。
- 磁碟檔案可以迅速地存取和改變內含資訊。

每個檔案均需賦予一名稱和型態（ type ）說明。當您想使用某一特定檔案內的資訊，您就得以“名稱”傳喚它使您使用。至於檔案的型態則是表示該檔案內含資訊的性質（種類）。唯有正確了

解各檔案的型態，才能下達正確的 **DOS** 指令。(關於"型態"的作用及意義，在第三章解說)

每一片磁碟均保存一份所有檔案的記錄表——檔案的名稱、型態、佔用空間大小及存放位置，我們常稱這份記錄表爲〝**目錄**〞(catalog)，因爲它在磁碟片扮演的角色就像是書籍的目錄表。圖2-3的方塊圖說明一目錄表記載著存放於 PERSONNEL 磁碟片各檔案的情形。各檔案的型態並不受限制。一目錄表可記載105個檔案的事項，然而在大多數應用下，一磁碟片很可能在目錄表爆滿之前，就可能被幾個檔案佔去所有磁片空間。

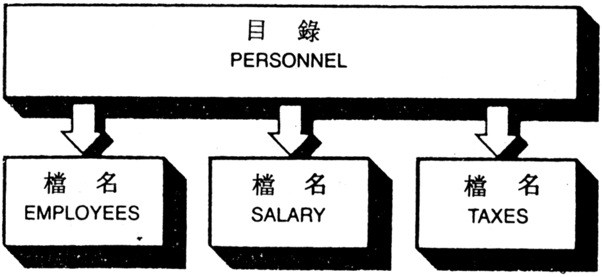


圖2-3　目錄表的檔案

檔案名稱

DOS 是利用"**檔案名稱**(filename)"來建立、辨別及管理各檔案。一檔案名稱(簡稱"檔名")的命名可多至30個字元(character)，但第一個字元必須是個英文字母，其他則是逗點以外的任何文字或符號。在一特定的磁碟片上，各檔名必須不重複(唯一名稱)。

下列是合法的檔案命名方式，請注意到"空白"也是檔名的一部份，其目的是使整個名稱自然、易懂。

- A SOMANMBULIST RUNNING RAMPANT 是最長的一個檔名範例。
- Z 是最短的一個檔名範例。
- A-1 DERFUL-NAME-3 是句點，字母及數字配合使用的一個範例。

假若您打入的檔案名稱其字元長度超過規定的 30 個，則DOS 會忽略多餘的字元。例如，檔名 TWO-SOMNAMBULISTS-RUNNING-RAMPANT 共有 33 個字元，則DOS 自動截短成 30 個字元的 TWO-SOMNAMBULISTS-RUNNING-RAMP ，它不會印出錯誤訊息。只要您沒有打錯字，而且該截短的名稱也是獨一無二時，您打入完整的 33 個字元檔名時，DOS 仍會認得它的！

下列是DOS 無法接受的檔案名稱命名法：

- 不要以數字爲檔名的開頭字元，例如 2HT2HL 是不對的。
- 不要以句點爲檔名的開頭字元，例如.PRINTER 。
- 不要在檔名內包含有逗點，例如 BOOP, BETTY 是無法接受的檔名。

若它是個不合法的檔案名稱，DOS 會顯示 SYNTAX ERROR 錯誤訊息。

卷 號

放置於磁碟片內各種資訊的集合體，在Apple 系統上稱呼它爲"**卷號**"（ Volume ）；"卷號"的由來是因爲在顯示其值時，它是個"數目（號碼）"，有如書籍（雜誌）的裝訂來命名慣例（第一卷、第二卷……）。利用卷號， DOS 在必要的時候可以確定放置於磁碟機內的工作磁碟片是否是正確的磁碟片。

卷號的授予時機是在 INIT 指令啓始化一空白磁片的期間，以便往後的資訊儲存。（關於 INIT 指令的完整格式會在第三章詳加解說）。即使您在啓始化一空白磁碟片時，用 INIT 指令特意選用“卷號”功能（ volume option ），然而，您下達的DOS 指令並不需加註卷號。（譯註，“卷號”無多大作用，它可能只是便利人們標示，在衆多磁片加以區別罷了。）不加註卷號的情況下，DOS 會假設該卷號是預定值 254 。

DOS指令

使用 DOS 指令就如同在程式語言中使用某些專用“字語”（ word ）的情況一樣，須遵循某些特定的規則。有一些規則可能是整體性，少數幾項規則只適用在某些情況的，而本章節是討論一般性的規則。

大多數的 DOS 指令不僅可自鍵盤打入，也可包含在程式中，不過本部份只介紹自鍵盤下達 DOS 指令的用法，至少在程式中使用 DOS 的詳細說明，請參考“DOS程式設計者手册”（本書的第二部份）。

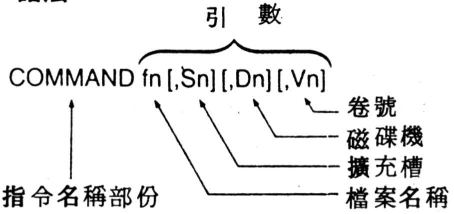
語　法

各個DOS指令的**語法**（ Syntex ）（“語法”可說是構成一完整指令的各部份排列次序及格式）在本書上是以下列的記號或簡字來表示的：

UPPERCASE	大寫字母的部份，表示這是個“眞實的”名稱，請忠實地打入此部份。
lowercase	小寫字母的部份，表示您可隨意打入的任何文字；例如一程式的名稱，您可隨意命名。
fn	表示須由您打入的檔案名稱。
[ ]	方括號內的事項，表示它是在一指令中的**引數**（argument），您可以加入該項，也可以不加入該項。但任何情況下，您絕不可打入方括號本身。
n	表示須由您打入的數目。

每一個 **DOS** 指令的各種可能的格式可以圖 2-4所示的一列敍述式來表示之。

語法



實例

```
SAVE ELEGANT.CODE,
D2
DELETE BOOKS, S6,
D1, V3
RUN WHIZBOOM, D1
```

圖 2-4

在圖 2-4 的 **COMMAND** 代表任何一個 **DOS** 指令之名稱（例如 **SAVE,DELETE** 或 **RUN**），而 fn,[,Sn]、[,Dn]和[,Vn]是指令的引數，加註方括弧的項目表示它是“選擇項”(option)，您可依據當時的情況“加入”它或者“從缺”。

選擇項

在方括弧內的引數稱之爲〝**選擇項**〞（ option ），表示它是可有可無的項目。若您用了某選擇項，其前的逗點是爲了與前一個引數分開以便辨認。在逗點的前後均可加入空白。請打入“大寫”的字母（S，D或V）表示您選用了某個選擇項，而“n”是您自取的某個數目。注意，方括號在本書是作爲“選擇項”的特殊記號，在您打入 DOS 指令時，絕不許打入方括號。

預設值

假若您在使用 DOS 指令時未明示各種選擇項，則 DOS 自行假設一值代入該選擇項，該值即稱之爲“**預設值**”（ default ），也就是在未明示某引數的內含值，系統即以此值當作其值。

在磁碟啓動過程中，DOS 對於使用的磁碟機代號，擴充槽及卷號均有各別的預設值，在正常情況下，其預設值如下：

- 磁碟機1（ drive 1 ）
- 擴充槽6（ slot 6 ）
- 卷號254（ volume 254 ）

除非您特別指定不同的磁碟機，擴充槽或卷號，DOS 會一直沿用這些系統預設值。

例如，在啓動之初，第一部磁碟機（ drive 1 ）是系統預設值（DOS 一定得自第一部磁碟機啓動之），而後，您下達了CATALOG指令，DOS 就會列出於第一部磁碟機內磁碟片的目錄。倘若您希望察看第二部磁碟機內磁碟片的目錄，您必須打入

C A T A L O G , D 2

一旦您使用了D 2這一選擇項，**DOS** 就一直假設第二部磁碟機是您往後使用的“預設”磁碟機，除非您再度改變D項引數（例如D 1）。

DOS 對於擴充槽號碼也是同樣的情況。擴充槽 6（假設**DOS**查出磁碟機控制界面卡是插在編號 6 的擴充槽）是預設的擴充槽，除非您告訴 **DOS** 使用別的擴充槽。

假若您打入如下的指令，

C A T A L O G , S 5

DOS 會查看編號 5 的擴充槽是否有磁碟控制界面卡，倘若確實接上該卡，**DOS** 將依令列出該界面卡第二部磁碟機（因爲您剛剛選擇了D 2項）內磁碟片的目錄。往後，一直沿用編號 5 的擴充槽。如果要改回先前“擴充槽 6”及“磁碟機 1”，您必須再下達有關於這二項的任何指令，例如， CATALOG,S6,D1. 。

（**註**：當您想更動系統的預設值，您必須很明確地告訴**DOS**新值。對於下一個指令的各個引數，這些新值會一直沿用下去。）

數　目

本書的語法說明式裡的小寫字母 n ，表示它是項“數目”，該數目是該引數的一部份。例如［，Dn］的記號表示您若選擇了該項，就必須以實際的數目代入之，告訴 **DOS** 磁碟機的編號，您可打入：

, D 1

或

, D 2

您可以用**整數**（ integer ，相當於自然數，不帶小數位數的數值）或**十六進制數目**（ hexadecimal number，利用 0 至 9 的數字

，以及A至F的文字表示其值，例如A 3相當於十進制的163。）來表示n的值。若您想利用十六進制數目，請參考“DOS 程式設計者手册”的說明。

引　數

fn，[,Sn]，[,Dn]和[,Vn]等四項引數(argument)分別表示您希望的檔案名稱，擴充槽編號、磁碟機代號及磁碟片卷號，組合各引數就可明確地指明檔案所處的“環境”。加註方括號的引數表示它是項“選擇項”(option　)。當您沒有明示選擇項的引數值時，DOS 會如上節所示自取預設值。至於上述各引數的排列次序是無關緊要的，例如 V36,D2,S6 的意義是與 S6,D2,V36 相同的。

檔案名稱：fn

在語法說明式裡的fn是代表“filename”(檔案名稱)，幾乎所有的 DOS 指令都“必須”有檔案名稱這一項引數。在這種情況下，您使用 DOS 指令時，必須以“檔案名稱”爲該指令的第一個引數。

假若目前的預設磁碟機正是您希望的磁碟機，那麼您只要打入指令名稱並後隨一個檔案名稱。例如您打入

```
SAVE PROGRAM
```

和

```
RUN OLD.HAT
```

DOS 會自動假設擴充槽、磁碟機和卷號的預設值。若預設磁碟機是第二部磁碟機(drive 2)，上述兩指令即是使用第二部磁碟機

內的磁碟片。利用預設值可以省時、省力，讀者諸君不妨多運用此法。注意，DOS 會一直認定指令的第一個引數是檔案名稱。

擴充槽編號：[,Sn]

本選擇項引數是用來明示 DOS 指令適用的某個擴充槽編號。當您使用本項時，請用 1 至 7 的數目實際代入 n 的位置，該數目即是往後的擴充槽預設編號，除非您再使用不同的數目改變它。

假若您不選用本引數，DOS 自動找出您最近設定過的擴充槽編號。若開機後使用 DOS 來作事均未選用擴充槽引數，那麼DOS 即認為擴充槽預設值是您啓動 DOS 時候所用的擴充槽號碼。

若您擁有一張以上的磁碟機控制界面卡或是同時有印表機控制界面卡在主機內時，您才有必要選用擴充槽引數。倘若您的主機配備四部磁碟機，使用兩張磁碟機控制界面卡，區別它們是絕對有需要的，方法如下：

- drive 1 : ,S6,D1
- drive 2 : ,S6,D2
- drive 3 : ,S5,D1
- drive 4 : ,S5,D2

本項引數對於印表機的選擇，或者插置於主機內各擴充槽資訊傳輸的裝置之選用亦同時適用。

（**什麼時候不需要說明擴充槽編號**：假若您只有一或兩部磁碟機，而且您的Apple電腦內也無其他週邊裝置界面卡，那麼在DOS 指令中無須加入擴充槽引數。只要您不特別說明一個不同的號碼，DOS 會一直認為目前的擴充槽即是啓動系統時候的擴充槽。）

磁碟機代號：[,Dn]

本磁碟機代號項目是可有可無的選擇項引數，目的是在告訴 DOS 您想使用的磁碟片是放在那一部磁碟機上。

當您使用本項時，請以1或2代入n的位置；倘若您省略磁碟機代號引數，DOS 就假設它是您最近用過的磁碟機。在開機後一直沒有特別明示磁碟機代號，那麼 DOS 認爲啓動系統的磁碟機編號即是目前的磁碟機代號，下列就是 DOS 指令中選用擴充槽及磁碟機引數的實例：

CATALOG D2	顯示第二部磁碟機內磁碟片的目錄，DOS 會以最近用到的擴充槽爲目前擴充槽號碼。
RUN WHIZBOOM ,S6 ,D1	自連接於編號6的擴充槽內磁碟機控制界面卡的第一部磁碟機讀出 WHIZBOOM 程式，並執行之。

（**什麼時候不需要說明磁碟機代號**：假若您只有一部磁碟機，同時也無其他週邊裝置界面卡插在Apple電腦內，那麼您沒有必要加註磁碟機代號，只要您不特別說明一個不同的號碼，那麼 DOS 會一直沿用啓動系統時所使用的磁碟機。）

卷號：[,Vn]

本卷號選擇項引數是用來明示使用中（或想使用的）的磁碟片編號。雖然您可以在大多數的 DOS 指令選用本項引數，但它其實是很少被使用的。

假若您希望加註卷號，請用1至254之間的數目代入n的位置。只要您選用某一個數目，那麼該數目即變成目前的預設卷號。如果您在使用 INIT 指令啓始化一空白磁碟片之時，並未選用不同的數目，那麼 DOS 就假設254是該磁片的卷號。

當您繼續使用其他指令時，不再選用卷號引數，或者認定卷號爲0，或者只打入 V 但並未後隨一個數目，那麼 DOS 會認爲記載在磁碟片上的卷號即是您希望的卷號。下列是使用卷號選擇項引數的例子：

LOAD DICE, V21　　DOS 會查驗在目前預設的磁碟機內磁碟片的編號是否是21，如果吻合，那麼DOS 會自磁碟片讀出 DICE 程式並載入記憶體內。

不正確的引數造成的錯誤訊息

假若您採用的各引數有錯誤的話，您可能會有下列一項或多項錯誤訊息：

- RANGE ERROR 訊息是告訴您使用大於 7 的擴充槽編號或者不是 1 也不是 2 的磁碟機代號。
- VOLUME MISMATCH 訊息是告訴您 DOS 找不著該卷號的磁碟片。您可以在 DOS 誤用另一片磁碟之前，打入有正確卷號的指令。
- I/O ERROR 訊息的出現有下列數種情況：

  也許是指定的磁碟機的門沒關好，請先關好它，再試一次指令。也許是磁碟機的信號綫鬆掉了；檢查信號綫的一端（擴充槽）是否已接上。

  也許是在指令的擴充槽上並未接上界面卡，倘若指定的擴充槽並無界面卡，那麼主機會停止不動，而且在記憶體的東西會漏失不見。在這種情況下，您只好重新啓動DOS。

第2章摘要

在本章中，您已學會DOS 如何動作，了解使用DOS 指令的語法（格式），同時也學得“檔案”、“卷號”彼此之間的關係以及相關於DOS 的細節。

當您啓動DOS時，您應該知道螢幕上會出現怎樣的訊息，以及DOS是如何與Apple電腦內其他程式配合作事。

3 使用磁碟

- ■CATALOG 指令 83
  - 了解目錄的意義 85
- ■拷貝用程式 87
  - Applesoft BASIC下的COPYA程式 87
  - Integer BASIC下的COPY程式 89
  - 拷貝過程中的錯誤訊息 89
- ■準備好可用的磁碟片 90
  - 啓始化的磁碟 91
  - 主磁碟 92
  - 歡迎詞程式 92
  - INIT指令 93
- ■決定擴充槽和磁碟機代號：SLOT＃程式 95
- ■第3章摘要 96

第3章 使用磁碟

本章的重點是討論如何管理磁碟，並說明如何獲得磁碟片內各檔案（名稱、性質及其長度的列表，如何複製整片磁碟，爲DOS準備好一磁碟片，以及決定目前擴充槽和磁碟機代號。這些課題涵蓋了第一章所述的內容並更加詳盡解說各指令意義。

CATALOG指令

磁碟片內容的一覽表習稱爲〝目錄〞（CATALOG）。當您下達 CATALOG 指令後，關於磁碟片內各檔案及其相關的統計數字（佔用的磁片空間之大小）會顯示出來。除了列舉各檔案名稱外，同時也標示某檔案是否已“上鎖”（ lock ），檔案的型態（ type ）以及檔案大小。

在磁碟片上的目錄是一連串的記錄，每一筆記錄均記載者檔案名稱、存放位置以及檔案的相關資料。磁碟的目錄總共預留105筆記錄項，然而常在填滿磁碟目錄之前，磁片的空間已全分配完畢。

當目錄各細項的數目已超過螢幕一次所能顯示的項目，您必須隨意按下一鍵，讓它繼續顯示其餘的部份。按空白鍵（SPACE，空間棒）是較好的習慣。若您想再查看目錄的前面部份，就得再下

達一次 **CATALOG** 指令。

列出目錄的指令其格式如下：

```
CATALOG [,Sn] [,Dn]
```

假若您加註了卷號引數在 CATALOG 指令中，DOS 會忽略不管。但假若您未選用擴充槽編號（ Sn ）和磁碟機代號（ Dn ）兩項引數，則DOS會自取系統的預設值（ default values ）——其值即是“最近”指定的擴充槽及磁碟機號碼（見第二章說明）。

提醒您：

- DOS 指令本身須以“大寫字母”打入。若您使用的是 Apple IIe 機型，請按下 [CAPS LOCK] 鍵“鎖”在“ON”（表示大寫）的情況。
- 在打入完整的指令，已更正錯誤並確認無誤後才按下 [RETURN] 鍵。

若您已在使用含有DOS的磁碟片，想查看在 SAMPLE PROGRAMS 磁碟片的目錄，請將 SAMPLE PROGRAMS 磁碟片放在“第一部磁碟機”（ drive 1 ）打入

[C][A][T][A][L][O][G][SPACE][D][1]

注意，在本指令中您無須使用逗號，因爲僅含一項引數。下達本指令後，螢幕上會顯示 SAMPLE PROGRAMS 磁碟目錄如圖 3-1 所示。請按下 [SPACE] 鍵繼續查看目錄的其餘部份。

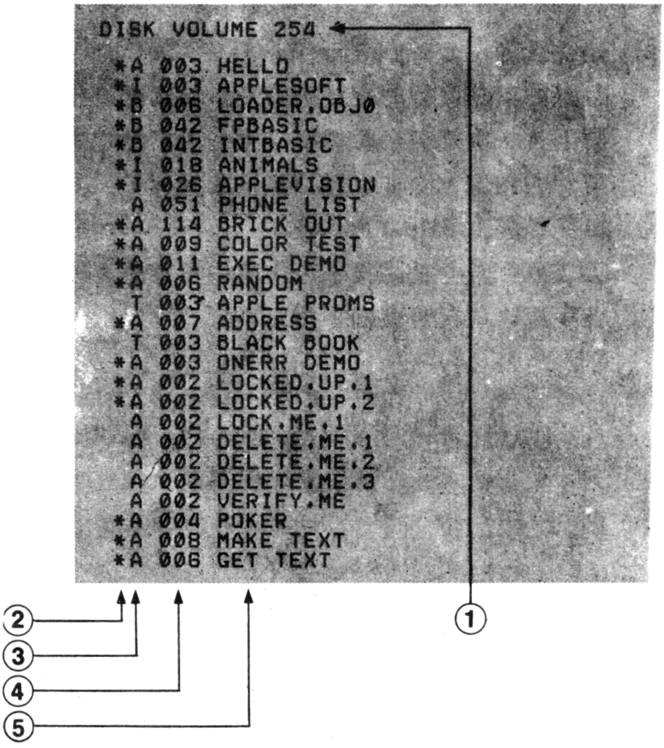


圖 3-1　SAMPLE　PROGRAMS 磁碟的目錄

了解目録的意義

下列是目錄內各部份的解說，標示的號碼是配合圖 3-1的說明。

① 卷號（Volume number）：
本數值是在磁碟片啓始化時設定的。它標示在目錄的前頭部份。

自目錄的左方開始對照參考，分別是

② 鎖住狀態（ Lock Status ）：
目錄的第一直行的記號是區別各檔案是否已“鎖住”。星號（ * ）即是表示該檔案已上鎖，已加以保護，您無法改變它的內容，除非您解除鎖住狀態。

③ 檔案型態（ File Type ）：
第二直行的一個英文字母，表明檔案所屬的型態。（註：各字母的代表意義如下所示：

檔案型態	**代碼**	**意義**
Applesofe BASIC	A	Applesofe BASIC ；以 SAVE 產生的檔案。
二進碼（ Binary ）	B	二進碼（機器碼）；以 BSAVE 產生的檔案（ * ）
Integer BASIC	I	Integer BASIC ；以 SAVE 產生的檔案（ * ）
文字檔（ Text ）	T	文字或文件；以 OPEN 產生的檔案
可重定位 (Relocatable)	R	組合語言（ + ）
內部檔案	S	爲未來的用途預留本代碼

（ * ）：參考 DOS 程式設計者手册

（ + ）：參考 6502 組合語言／DOS 工具程式手册（Assembler/DOS Tool Kit

④ 檔案大小（ File Size ）.
第三直行是顯示該檔案佔用的磁區（ Sector ）數量。（註：假若某檔案的實際佔用磁區大小超過 255，則顯示的數值是超過 255 的部份後再自零起算的數值。）

⑤ 檔案名稱（ Filename ）
最後一部份是檔案名稱。

拷貝用程式

複製整片磁碟的程式，在 DOS 系統磁碟片有兩個極為相似的拷貝用程式。至於該使用那一種，是由您使用的 BASIC 類型（Applesoft 或 Integer ）來決定的。COPYA 程式適用於 Applesoft BASIC 情況，而 COPY 程式是適用於 Integer BASIC 。

拷貝用程式（ COPY 或 COPYA ）在執行之初，會向您詢問擴充槽和磁碟機代號，若您希望採用系統的預設值，就直接按下 RETURN 鍵，否則就打入您希望的任何號碼。

複製整片磁碟的操作過程，在第一章已經逐步講解，而本章節是概略性的回顧，提供讀者精簡式的重點整理。（有益的建議：我們特別建議您最好複製一片以上的 SYSTEM MASTER 磁碟片，請將原版的 SYSTEM MASTER 保存於安全的地方，操作當中請使用拷貝版磁碟片。）

Applesoft BASIC 下的COPYA程式

若您使用的是Applesoft BASIC ，請用COPYA程式複製一整片磁碟至另一片空白磁碟。注意，您是從系統提示字元]，獲知您是使用 Applesoft BASIC 。

在 SYSTEM MASTER 磁碟片，執行COPYA程式，請打入

R U N SPACE C O P Y A , D 1

一旦程式已讀入記憶體內，螢幕上會顯示 APPLE DISK DUPLICATION PROGRAM 以及對應於原始的（ original ）和複製副本的（ duplicate ）磁碟的預設擴充槽和磁碟機代號。在本文中的“原始磁碟”（ original disk ）是意味著複製過程中的“來源”（ source ），而“副本磁碟”（ duplicate disk ）即是相對的“目的地”磁碟（ destination ）。

假若您同意螢幕上顯示的預設值，直接按 RETURN 鍵便可。若想改變這些預設值，請打入您希望的數目，並在一切準備就緒後，按下 RETURN 鍵，立即執行複製的工作。（註，您若不知目前磁碟機所處的擴充槽編號，可利用稍後會提到的 **SLOT#** 指令，或者直接掀起Apple電腦的上蓋，看看磁碟機控制界面卡是放在那一個擴充槽上。）

假若您只使用一部磁碟機進行複製磁碟，那麼在進行過程中，本程式會適時的提醒您抽換原始磁碟片和副本磁碟片。如果您擁有兩部（或以上）磁碟機時，可以將原始磁碟片放在第一部磁碟機（ drive 1 ），把空白磁碟片（即當作“副本磁碟”）放在第二部磁碟機（ drive 2 ），便可省掉交互抽換磁碟片的痲煩，快速的完成複製整片磁碟的工作。

在程式執行中，螢幕上會不時的顯示出 INITIALIZING, WRITING, 和 READING. 等訊息。在拷貝完成後，它會詢問您是否想再作一次複製工作，若您打入的是 N，表示不再繼續，程式自動中止並讓提示字元] 重現於螢幕上。

若您想再多作一次複製磁碟工作，請打入 Y，程式將重頭再執行一次。如果您想得到一步一步執行的操作要領說明，請參閱第一章的說明（包括一部磁碟機和兩部磁碟機情況）

在進行拷貝之前，爲預防萬一，請先爲“原始磁碟”加上“寫入保護”貼紙（覆蓋住磁碟片“可寫缺口”），可以避免無意中毁去該磁碟片上資料。

Integer BASIC 下的 COPY 程式

若系統提示字元是>記號，表示 Integer BASIC 正等待著您！在此情況下，您可以使用COPY程式複製整片磁碟片至另一空白磁碟片。

在 SYSTEM MASTER 磁碟片，執行 COPY 程式，請打入

R U N SPACE C O P Y , D 1

而後，請使用者遵循螢幕上的訊息打入適當的數目或放入磁碟。COPY 程式的動作情形與COPYA極爲相近，請讀者參考上一節的詳細說明。

若您想得到更詳細的逐步解說及顯示訊息（包括一部或兩部磁碟機的情況），請參考第一章的說明。

拷貝過程中的錯誤訊息

依據您使用的拷貝程式（COPYA或COPY）而定，若是 ***** UNABLE TO READ ***** （COPYA）或者 I/O ERROR STOPPED AT（列號）（COPY）訊息，表示您可能沒將磁碟片放入磁碟機內，或者未完全插入磁碟片，甚至是磁碟機的門沒關好，或是打入的擴充槽及磁碟機代號弄錯了。

依據您使用的拷貝程式（COPYA或COPY）而定，若是 ***** UNABLE TO WRITE ***** （COPYA）或者 I/O ERROR STOPPED AT（列號）（COPY）訊息，表示您的副本磁碟沒有放入磁碟機內，或是將可寫缺口貼上了保護貼紙，或是磁碟機的門沒關好，或是打入的擴充槽及磁碟機代號弄錯了。

假若COPYA程式無法讀取原始磁碟資料，或是無法寫入資料

至副本磁碟，將出現 DO YOU WISH TO MAKE ANOTHER 'COPY?訊息，讓您選擇是否再度嘗試，若打入[N]並按[RETURN]鍵，則回到 BASIC 控制之下。若COPY 程式已顯示 I/O ERROR 錯誤訊息，則直接回至 BASIC 之下。

如果COPY程式無法成功地啓始化一副本磁碟，則螢幕上顯示的是 I/O ERROR及 Integer BASIC提示字元。若COPY無法進行資訊的讀寫及轉移動作，顯示出" DO YOU WISH TO MAKE ANOTHER COPY?"時，您打入[N]並按下[RETURN]鍵時，則 Integer BASIC 提示字元重現於螢幕上。

在您再次使用拷貝程式之前，請查驗 SYSTEM MASTER 磁碟，副本磁碟及磁碟片上的可寫缺口、擴充槽和磁碟機，看看是應保持原狀或稍作變動。

附註：另有一程式可以複製整片磁碟，不過它的速度比較慢些。請使用 FILEM 程式。 FILEM 程式的最大用途是在拷貝單一檔案，或者相近名稱的一些檔案，例如 FOO, FOO1 和 FOO2 。關於FILEM 詳情，請參考第 4 章。

準備好可用的磁碟片

從經銷商購得的新磁碟上是沒有任何有意義的資訊。若想配合 DOS 使用該磁碟片，您必須先存入一些特殊的資訊在內。將此種資訊存入磁碟片的動作，稱爲**啓始化**（ Initializing ）或**規劃格式**（ formatting ）。

您可以準備好一磁碟片以爲未來接收（儲存）資訊之用；您可以很單純地啓始化一磁碟片，或者更進一步成爲一" **主磁碟** "（ master disk ）。

啓始化的磁碟

您是利用 INIT 指令啓始化一磁碟，該指令將磁碟的表面規劃成數個**磁軌**（ track ）和**磁區**（ Sector ，**或稱磁扇** ）以便資訊儲存之用，同時也將一份 DOS 程式及歡迎詞程式存入磁碟內。

任何已作過啓始化步驟的磁碟，其操作能力是與系統記憶體有關的：當初規劃磁碟的系統記憶體決定了可使用該磁碟片的系統記憶體之最小需求。例如，若在 32K系統下規劃某一磁碟，則往後受到的限制如下所述。

- 它只能在 32K或 32K以上記憶體的系統上使用。
- 倘若某程式連同 DOS 本身（ DOS 佔用 10.5K記憶體 ）總共用去 32K以上的記憶體，則該程式將無法順利地被執行。（ 換句話說，該程式最多只能佔用 21.5K的記憶體，否則無法被執行。 ）

假若您是天天操作相同記憶容量的 Apple 電腦，那麼磁碟片的啓始化及儲存資訊工作，對您而言將是順心愉快，不會有任何麻煩事。然而，您若使用一磁碟片（ 也許是從朋友那邊借來的 ）於它允許的記憶體容量之下的 Apple 系統，則可能會有 OUT OF MEMORY 錯誤訊息。

順便一提： 在某些先前出版的手册裡，稱呼這一類的磁碟片爲“副磁碟 ”（ 或“ 奴磁碟 ” Slave disk ），因爲在啓始化的過程中，當時的系統限制了它往後適用的系統種類。（ 譯註：“ 奴 ”（ slave ）一字意味著“ 無法自主 ”或“ 受某種限制 ”。 ）

主磁碟

所謂的“主磁碟”(master disk)是表示該磁碟片具有自動調節以配合使用中系統記憶體的DOS 程式。換句話說，每回啓動主磁碟時，DOS 會儘可能將它本身載入記憶體中較高位址(譯註，記憶體容量越大，表示可使用到的位址值(address)也越高)，以便留下較多的記憶體空間給您使用。

主磁碟，相對於副磁碟而言，表示它是與記憶體大小無關的，其程式可在不同記憶體容量的系統中運轉。SYSTEM MASTER 即是“主磁碟”，它可在任何Apple電腦中使用。

假若您有兩部或多部不同記憶體容量的Apple電腦，想讓已作過啓始化步驟的磁碟轉換成“主磁碟”，請執行MASTER 程式。詳細操作說明，請參閱“DOS 程式設計者手册”部份。

歡迎詞程式

所有利用 DOS 作過規劃格式(啓始化)步驟的磁碟片都存有一個“歡迎詞程式”(greeting program)。該程式的內容在磁碟片被啓動後會自動顯示於螢幕上，它可以是“ INITIALIZED DISK CREATED ON MY BIRTHDAY ”簡單句子或是任何人創造的生動句子或者操作說明書。

若您想省事些，您可以複製既有的磁碟內歡迎詞程式到另一磁碟片，或者重新寫個較簡單的。任何形式的資訊均可放置歡迎詞程式內，然而它的性質是在於啓動磁碟後顯示一些人們能立即理解的訊息，最好其內容包含有

- 啓始化磁碟的日期。

- 當時使用的系統（記憶體容量）
- 友善的開機後“歡迎詞”文句

您可以自行設計有自已風格的歡迎詞程式，方法如下：首先下達 **NEW** 指令清除目前記憶體的內含資訊，而後打入可顯示某種訊息的 **BASIC** 程式，例如：

```
10 PRINT "MERRY CHRISTMAS!"
20 PRINT "64K DISK INITIALIZED 25 DECEMBER 1983"
30 END
```

附帶一提的是，您可以使用任何名稱為“歡迎詞程式”檔案名稱。由於它也如同一般的檔案，其名稱也會在顯示的目錄當中，為了“與衆（檔案）不同”而又為了可讓大家“一致認同”該檔名是最恰當的名稱，本書建議讀者使用 **HELLO** 為歡迎詞程式之檔案名稱。

一旦您設計好了歡迎詞程式，請使用 **INIT** 指令（例如 INIT HELLO,D1）將該程式存入磁碟片並進行啓始化磁碟的處理過程。

INIT 指令

INIT 指令是用來規劃（啓始化）一磁碟片來配合 **DOS** 的使用。本指令會將磁碟片的表面規劃成數十個磁軌和磁區，並將 **DOS** 本身及一歡迎詞程式存入磁碟內。

當啓始化工作完成後，**DOS** 並未顯示任何通告結束的訊息，只是讓提示符號重現於螢幕而已。而後，您可以使用此已啓始化磁碟來啓動系統，**DOS** 也能在此磁碟上讀取和寫入資訊。

當一啓始化磁碟已啓動，**DOS** 會查驗 **INIT** 指令所指引到的程式並且立即執行它。大多數情況下，是指引到一個歡迎詞程式，所以啓動磁碟時會顯示一些訊息。某些情況下，它是指引到其他程

式（可能是電玩遊戲或應用程式），該程式也會自動執行的。若一程式能在磁碟啓動時自動執行，常稱它是**轉鑰程式**（ turnkey program ）（譯註：感覺上有如扭開開關即刻有作用，詳情請見“DOS 程式設計者手册”）

您可以反覆地“回收”已啓始化磁碟。只要您不再希望保存某一磁碟內的資訊，您可以再度啓始化該磁碟片。若進行啓始化工作，會清除其內的資訊回復一片可用的“新”磁碟。（**警告**：在啓始化一磁碟片時，要確定儲存於磁碟的資訊已不再有價值，方可下達 INIT 指令。）

您只能在“立即執行模式”（ immediate execution ）下使用 INIT 指令，換句話說，您只能從鍵盤上打入本指令（無法在 BASIC 程式含有本指令）。

INIT 指令的語法如下：

```
INIT fn [,Sn] [,Dn] [,Vn]
```

此處的 fn 即代表“檔案名稱”（ filename ），也即是目前存在記憶體內的程式名稱，DOS 會將該程式存放在指定的磁碟。例如，第一章的三列 BASIC 顯示聖誕快樂句子的程式若已存在記憶體內，您可以打入

[I][N][I][T][SPACE][H][E][L][L][O]

DOS 會將該程式以 HELLO 名稱儲存於預設磁碟機內磁碟。假若目前記憶體並無任何程式在內，DOS 則自行產生一空白檔案儲放於磁碟上。

在本指令的擴充槽、磁碟機代號及卷號等三項引數均屬於可有可無的選擇項（ option ）。DOS 會沿用先前 DOS 指令的該三項引數值。假若您不確知目前的預設值，那麼就自行打入您希望的數值。當您省略卷號引數時，或者設定它是 0 ，或是打入[V]但不接上

任何數目， DOS 會假設卷號是 254。若您打入一確定的數值，則 DOS 即以該值爲目前磁片的卷號。

INIT 指令是用來設定卷號的唯一 DOS 指令，假若您使用了其他 DOS 指令之時也加註卷號引數，則指明的卷號必須與該磁片在使用 INIT 指令設定的卷號相吻合。

關於啓始化一磁碟過程的逐步解說，請參考第 1 章的說明。同時，編者建議您，若您有數片空白磁碟，不妨在同一時間內同時規劃它們的格式，以備未來之需。

決定擴充槽和磁碟機代號： SLOT #程式

SLOT #程式可以回報目前使用的磁碟機所在的擴充槽編號及磁碟機代號，也即是顯示它們的預設值。

某些程式（如拷貝用程式）會向您詢問擴充槽及磁碟機號碼。而某些指令，尤其是 INIT 指令，若您不確知目前的預設值，很可能會危害到磁碟的資訊。

爲了驗證目前的擴充槽編號及磁碟機代號，您可以執行 SYSTEM MASTER 磁碟片內的SLOT #程式；請將 SYSTEM MASTER 磁碟放在第一部磁碟機，打入

R U N SPACE S L O T SHIFT #

該程式將會回報如下的訊息：

```
CURRENT SLOT IS 6 AND DRIVE IS 1
```

第3章摘要

指　　令

CATALOG [,Sn] [,Dn]	顯示指定的磁碟內所有檔案的一覽表（目錄），包含有檔案名稱、佔用磁片空間之大小，檔案型態，檔案是否鎖住的狀態。
INIT fn [,Sn] [,Dn] [,Vn]	目的是預備好一磁碟片以為儲存資訊之用。 INIT 指令會將 DOS 程式及歡迎詞程式存入磁碟內。

程　　式

COPYA	在使用 Applesoft BASIC 情況下，複製一整片磁碟。在執行之前把副本磁碟啓始化（規劃格式）是不必要的事前準備。
COPY	類似 COPYA 程式之功能。差異是它在 Integer BASIC 情況下執行之。
SLOT#	回報目前擴充槽及磁碟機的預設值。

4 使用檔案

■ RENAME指令　99

● 實　例　101

■ LOCK指令　102

● 實　例　103

■ UNLOCK指令　103

● 實　例　103

■ DELETE指令　104

● 實　例　104

■ VERIFY指令　105

● 實　例　106

■ FILEM程式　106

● 指定擴充槽編號和磁碟機代號　107

● 利用“通配字元”指定檔案名稱　108

● 選擇FILEM的功能項　110

(1)COPY FILE（複製檔案）　110

(2)CATALOG（目錄）　114

(3)SPACE ON DISK（剩餘的磁碟空間）　114

(4)UNLOCK FILES（檔案解鎖）　115

(5)LOCK FILES（檔案鎖定）　116

(6)DELETE FILES（刪除檔案）　118

(7)RESET SLOT & DRIVE（重定擴充槽及磁碟機代號）　118

(8)VERIFY FILES（查證檔案內容）　119

(9) QUIT（結束FILEM程式） 119
• FILEM顯示的訊息 120
■第 4 章摘要 121

第4章

使用檔案

本章的目的是指導您追踪記錄和管理磁碟片的各檔案，您可以利用 DOS 指令來改變檔案的名稱，在磁碟片預留更多的空間，或為預防檔案的內含資訊無意中毀損加上某種保護措施，而且也能夠查證磁碟內某一檔案是否可以正確地讀取內含資訊。

本章後頭提到的 FILEM 程式會讓您在同一段時間內操作磁碟片數個或所有檔案。同時，本章會詳細解說每一個指令並以一個例子示範指令的用法。當您想嘗試測試這些例子時，請最好以拷貝版的 SAMPLE PROGRAMS 磁碟片作為練習之用，保存“正本”磁碟於安全地方；而後在您學習DOS指令之後，別人想瞭解 DOS 指令操作時，完整的示範用程式仍可自正本磁碟再拷貝一份。

RENAME 指令

RENAME 指令可以將磁碟內任一檔案的檔名重新命名，不論它的型態是 BASIC 檔案，文字檔（ text file ）或二進制機器碼檔案（ binary file ）。

RENAME 指令的語法如下：

RENAME fn1,fn2 [,Sn] [,Dn] [,Vn]

RENAME 指令會將取名爲 fn1 的檔案更名爲 fn2。名稱爲 fn1的檔案必須是磁碟片中一既存的檔案，而且"尚未上鎖"(unlock)。新名稱 fn2 必須是磁碟片中不重複的唯一名稱。

本指令中的擴充槽、磁碟機和卷號等三項引數均是可有可無的選擇項(option)。在您不願沿用目前的預設值情況下，您才有必要加上各引數。這些引數的意義已在第 2 章介紹過。

假若您指定的 fn1 檔案並不存在於磁碟內，則會顯示 FILE NOT FOUND 錯誤訊息。若該檔案已鎖定，則出現的是 FILE LOCKED 錯誤訊息。

順便一提，另有一法可將檔案加以更名，方法如下：先將該檔案載入記憶體，而後以一新的檔案名稱儲存該檔案於磁碟上。

注意，一旦 RENAME 指令被執行時，DOS 並不負責查驗第二個檔名(fn2)是否已存在於磁碟，當 DOS 執行 RENAME 指令過後，會有相同檔名(fn2)的兩個檔案。

當一磁碟內有多個同名的檔案時， DOS 祇會"認出"第一個檔案，其餘的 DOS 無法加以存取資訊。例如，若一磁碟目錄如下所示。

```
HELLO
LETTERS
SORT
PROGRAM.1
```

您下達了更名指令

```
RENAME PROGRAM.1,SORT
```

新的目錄將如下所示：

```
HELLO
LETTERS
SORT
SORT
```

除非您將第一個 SORT 檔案再度更名，否則 DOS 無法使用到第二個 SORT 檔案。請選用一“唯一的”新名稱將第一個檔案更名，如此做將可以存取命名爲 SORT 的第二個檔案。爲了防止上述重名的現象，請在更名之前先行檢查磁碟的目錄。

實　例

1. 將 SAMPLE PROGRAMS 磁碟片放入第一部磁碟機。查驗該磁碟的目錄，請打入

   CATALOG,S6,D1

   按下 RETURN 鍵後，螢幕上會顯示一長列的檔案名稱。本指令選用了擴充槽（S 6）及磁碟機（D 1）等引數。假設該目錄包含了檔案 HELLO, ANIMALS, BRICK OUT 和 PHONE LIST。

   按下空白 SPACE 鍵，可以繼續顯示目錄的其餘動作。
2. 將 PHONE LIST（相當於 fn1）檔案更名爲DIRECTORY（相當於 fn2）。請打入

   RENAME SPACE PHONE SPACE LIST ,DIRECTORY

   按下 RETURN 鍵後，DOS 執行檔案更名的動作。
3. 現在，再度下達CATALOG指令查驗目錄，以確定其結果是否

如您所想的。記得，要按 SPACE 鍵查看目錄的其餘部份。

4. 再將檔名更名，還原成原先的名稱。指令如下：

R E N A M E SPACE D I R E C T O R Y
, P H O N E SPACE L I S T

LOCK 指令

LOCK指令可將檔案“上鎖”（lock），保護該檔案以避免無意中的更名、刪除或變動內容。不過，已上鎖的檔案照常可以被載入記憶體並執行之。

若試圖將一上鎖的檔案更名，刪除或變動其內含資訊，將會回應 FILE LOCKED 錯誤訊息。欲改變已上鎖檔案的內容，您首先必須下達UNLOCK指令以解除鎖定狀態。

已上鎖（鎖定）的檔案，在顯示的目錄部份會有特殊記號（星號加註在檔案型態之左方）。

順便一提：LOCK指令每回只對一個檔案有效。假若想保護整片磁碟，最簡單的方法是將磁碟片的“可寫缺口”(write-enable notch）覆蓋一張防止寫入的（銀質或不透光的）貼紙。

本指令的語法如下：

LOCK fn [,Sn] [,Dn] [,Vn]

指令中的 fn 代表您希望鎖定的檔案名稱。至於擴充槽（Sn），磁碟機（Dn）及卷號（Vn）均是選擇項引數。只在您不希望沿用目前的引數預設值時，您才有必要加上上述各引數。

實 例

1. 假設 SAMPLE PROGRAMS 磁碟已放在第一部磁碟機，利用CATALOG指令顯示其目錄，注意到LOCK. ME. 1檔案並無一星號，表示它並未鎖定。
2. 鎖定LOCK. ME. 1檔案，請打入

   L O C K SPACE L O C K . M E . 1

3. 利用CATALOG指令來查驗LOCK. ME. 1檔案已加上保護措施。注意到目錄中的該檔案部份已加註有星號。

UNLOCK 指令

UNLOCK 指令是用來解除檔案鎖定的狀態。它撤除對檔案的保護，以便讓您進行刪除、更名和變動其內容的動作。UNLOCK 指令每回只能對一個檔案發生效用。

本指令的語法如下：

UNLOCK fn [,Sn] [,Dn] [,Vn]

指令中的 fn 代表您希望解除鎖定狀態的檔案名稱。至於擴充槽（Sn），磁碟機（Dn）及卷號（Vn）均是選擇項引數。只在您不希望沿用目前的引數預設值時，您才有必要加上各引數。

實 例

1. 將 SAMPLE PROGRAMS 磁碟片放在第一部磁碟機，下達

CATALOG指令以查詢該磁碟的目錄。請注意到 LOCKED. UP. 1 和LOCKED. UP. 2檔案已加註星號，表示它們已“鎖定”。

2. 解除LOCKED. UP. 1 檔案鎖定狀態，請打入

U N L O C K SPACE L O C K E D . U P . 1

3. 再度使用CATALOG指令查看目錄，以驗證該檔案已解除鎖定狀態。該檔案的目錄部份（在其檔案型態之左）已無星號。

DELETE指令

DELETE 指令是用來刪除磁碟內某一檔案，其目的是去除某些檔案的老舊版本，騰出一些空間來容納新檔案。

本指令的語法如下：

DELETE fn [,Sn] [,Dn] [,Vn]

本指令標示的 fn 必須是個未鎖定的檔案（亦即是目錄部份並無星號加註於該檔案之前）。假若該檔案並不存在，下達本指令後會回應 FILE NOT FOUND 錯誤訊息。若該檔案已被設定鎖住狀態，則回應 FILE LOCKED 錯誤訊息。至於擴充槽（Sn）、磁碟機（Dn）及卷號（Vn）等引數是可有可無的選擇項。只在您不想沿用目前的各引數預設值下，您才有必要選用這些引數。各引數的意義，請參考第 2 章說明。

實 例

1. 將 SAMPLE PROGRAMS 磁碟放在第一部磁碟機，下達

CATALOG指令以查詢該磁碟的目錄。目錄中包含有DELE-TE.ME.1檔案。

2. 刪除 DELETE.ME.1 檔案，請打入

D E L E T E SPACE D E L E T E . M E . 1

3. 再度使用CATALOG指令查看目錄，以驗證目錄已不再含有 DELETE.ME.1 檔案的記錄，證實已永久刪除該檔案。

VERIFY指令

VERIFY指令是用來證實（查證）某一檔案已正確地寫入磁碟內，同時也能讓 DOS 讀取其內含資訊。若檔案已被查證無誤，則表示您可以放心地假設資訊已正確地儲存於磁碟上，可在您需要的時候讀取它。

若 DOS 能正確地讀取檔案內容，它會顯示提示字元於螢幕上，否則顯示 I/O ERROR 錯誤訊息以表示無法讀取該檔案（該檔案可能已毀損或曾錯誤地寫入資訊）。

爲了查證某檔案可讀取，DOS 會將該檔案內含資訊讀取出來並轉存至目前尚未使用的記憶體部份。本指令並不會影響已存在於記憶體的程式。

本指令的語法如下：

```
VERIFY fn [,Sn] [,Dn] [,Vn]
```

任何型態的檔案均可加以查證。若指名的檔案（fn）不存在於磁碟，則顯示 FILE NOT FOUND 錯誤訊息。至於擴充槽（Sn）、磁碟機（Dn）及卷號（Vn）等引數是可有可無的選擇項。只在您不想沿用目前的各引數預設值下，您才有必要選用這些引數。

實 例

1. 將 SAMPLE PROGRAMS 磁碟片放在第一部磁碟機，下達 CATALOG指令以查詢該磁碟的目錄。目錄部份應包含有 VERIFY.ME 檔案。
2. 查證該擋案，請打入

V E R I F Y SPACE V E R I F Y . M E

記住，若 DOS 能讀取該檔案，並無特殊的訊息顯示出來。

FILEM程式

到目前爲止，本章已介紹 DOS 指令讓您每回下達一指令時，可以更動一檔案名稱、鎖定一檔案、删除一檔案或者查證一磁碟。在 SYSTEM MASTER 磁碟的 FILEM 程式可以讓您執行相同的動作，但做的事情更多更快。它可以針對單一的檔案，或者同一時間內針對“一些”檔案。

要使用 FILEM 程式，請先將 SYSTEM MASTER 磁碟放在第一部磁碟機，而後下達 RUN指令，如下所示：

R U N SPACE F I L E M , D 1

在 EXECUTING FID 簡短訊息出現後，您會在螢幕上看到如下的功能表畫面（圖 4-1）：

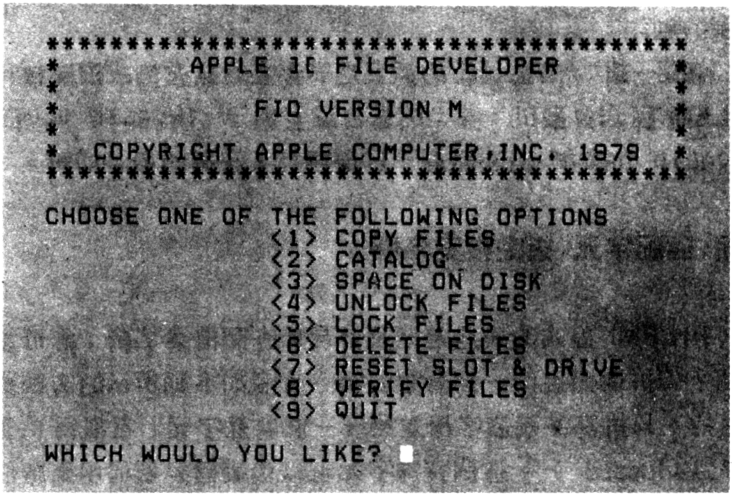


圖 4-1　FILEM功能表畫面

若要選擇某個指令，請打入標示該指令＜＞符號內的數目，而後按下 RETURN 鍵。只要打入該數目，不必打入＜或＞記號。（附註：FILEM 程式叫用了機器碼程式FID，故圖 4-1的畫面是標示著FID字樣。）

指定擴充槽編號和磁碟機代號

在“第一次”執行 FILEM 程式時，當您選擇了1，2，3，4，5，6或8等任一項動作，會先顯示 SOURCE SLOT? 問句，緊跟著的是 DRIVE? 訊息。SOURCE 表示“來源”，亦即是您希望讀取資訊的磁碟機所在環境。

從某一項轉換至另一項時（在1，2，3，4，5，6和8項之間），仍然會顯示如上的擴充槽及磁碟機提示字樣。假若您重複選用某項功能，本程式會沿用您已經指定的數值（擴充槽及磁碟機）。倘若您想選用某項功能，但想改變這些預設值，您必須執行第

7 項的動作。

順便一提： 若您給定的數目是不存在的擴充槽或磁碟機代號，FILEM 程式會發出“嗶”的警告聲並顯示 INVALID SLOT 或 INVALID DRIVE 錯誤訊息。只要再打入正確的數目即可。

利用“通配字元”指定檔案名稱

FILEM 程式的一些選擇項會向您查問檔案名稱，您可以一個一個字母拼成完整的一個檔案名稱，或者如本節所示的方法描述檔案名稱：只拼寫某幾個字母並夾雜一個特殊字元（常稱爲“**通配字元**” Wildcard ）。這個特殊字元可以代表檔案名稱中對應位置的一串字元（零個或多個字元）。這個“通配字元”是“等號”（＝）。（譯註：在撲克牌遊戲中，大、小鬼的紙牌有時可當作“任意花色（wild）的紙牌（card）”。本處取其含意爲“可隨意匹配”的一個記號，它可代表“任意的”文字、數目及符號）

例如，A＝S的表示法是代表以A字母開頭並以S字元結尾的任何檔案名稱。下述名稱通通合乎A＝S表示法的檔案名稱：

- AS ＝符號代表“無”任何字元在內。
- ACTS ＝符號代表C T。
- ACCESS ＝符號代表CCES
- ALLS WELL THAT ENDS ＝符號代表 LLS WELL THAT END

一旦您引用了“通配字元”於檔名之中，FILEM 程式將會顯示 DO YOU WANT PROMPTING? 訊息以詢問您是否需要“複查”的提示訊息。若您希望 FILEM 程式立即“靜悄悄”地完成合乎該檔名規格（filename specification）所有檔案動作，請打入 N 並按下 RETURN 鍵以表示不需再顯示“檔名”的提示訊息。

假若您希望於執行中再次複查指令中的檔案名稱，請打入 Y 並按下 RETURN 鍵。而後 FILEM 程式會在對各檔案有所動作之前顯示各個檔案名稱，若您確定該檔案名稱是正確的，請打入 Y 並按下 RETURN 鍵。若希望跳過該檔案不作任何事，則打入 N 並按下 RETURN 鍵，螢幕會顯示 CANCELLED 訊息告訴您它已跳過該檔案了。

建議您，在您處理完畢磁碟內所有您希望的檔案之後，您可以打入 Q 並按下 RETURN 鍵，強制回復功能表畫面並取消選用的指令動作。

圖 4-2 是利用"通配字元"來處理各組的檔案。

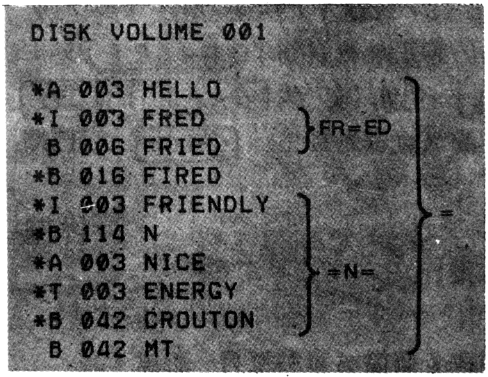


圖 4-2　利用通配字元代表各組檔案名稱

- FR=ED　代表所有以 FR 開頭並以 ED 結尾的檔案名稱。至於 FIRED 並不合乎本規格，因爲檔名的第二個字元是 I 不是 R。同理，FRIENDLY 檔名，其結尾是 LY 不是 ED。
- =N=　代表檔名中內含一個 N 字母的所有檔案。NICE 名稱中第一個字母即是 N。ENERGY 和 FRIENDLY 檔名內含一個 N 字，而 CROUTON 是以 N 結尾，也合乎規格

。甚至於單字母的N檔名也是允許的。

- = 代表所有檔案。在 FILEM 程式中單獨使用一個通配字元作爲檔名，即代表在選用的指令項是針對“所有”的磁碟片內檔案。

選擇 FILEM 的功能項

在 FILEM 程式內八項功能的其中六項相當於您曾經針對單一檔案處理的各種 DOS 指令，然而在 FILEM 程式中使用含有“通配字元”的檔名表示法可以一次處理數個檔案。

當 FILEM 已完成某項功能時，會顯示 PRESS ANY KEY TO CONTINUE 的訊息；此訊息的眞正意義是除了 RESET，SHIFT，CONTROL，(open-apple)，(solid-apple)，或 CAPS LOCK 之外的任何一個字鍵，不過，編者建議您按下 SPACE 空間棒（空白鍵）。

〈1〉 *COPY FILES*（複製檔案）

本選擇項自一磁碟內一個或多個檔案複製至另一磁碟。

注意，FILEM 程式與 COPYA 或 COPY 拷貝用程式的不同點，是它並不作副本磁碟（亦即是目的磁碟 destination disk）的啓始化（規劃格式的）動作。倘若您想利用 FILEM 來作多份拷貝版磁碟，請在執行本程式之前先行啓始化數片磁碟。關於啓始化磁碟的 INIT 指令，請參考第 3 章說明。

在 FILEM 執行第一項選擇項時，它會詢問“原始磁碟”所在的擴充槽及磁碟機代號。先顯示 SOURCE SLOT? 而後是 DRIVE? 問句。SOURCE 意味著您希望讀取資訊的磁碟。

隨後，它會詢問“目的磁碟”所在的擴充槽及磁碟機代號。螢

幕上顯示 DESTINATION SLOT? 而後是 DRIVE? 問句。DESTINATION 意味著您希望能接收（寫入）資訊的磁碟。

- 假若您僅有一部磁碟機連接至主機上，原始磁碟及目的磁碟所在的擴充槽及磁碟機代號是相同的。FILEM 程式會適時的通知您插入原始磁碟或目的磁碟至磁碟機內。
- 假若您有二部磁碟機的話，擴充槽編號對於兩者而言是相同的，但對應的磁碟機代號不同。FILEM 程式僅簡短的顯示 INSERT DISKS 訊息。

在 FILEM 已知悉擴充槽及磁碟機代號後，您會看到FILENAME? 訊息，要求您鍵入希望複製的檔案其名稱。下例是在兩部磁碟機情況下使用 FILEM 程式的操作解說。您應將SYSTEM MASTER 磁碟放在第一部磁碟機。假若螢幕不是 FILEM 功能表畫面，請打入如下的RUN 指令以啓動本程式

[R][U][N][SPACE][F][I][L][E][M][,][D][1]

操作程序		**結　果**
1.	自功能表中選擇第一項，打入 [1] 並按下[RETURN]鍵。	螢幕上顯示 COPY FILES 字樣，並印出 SOURCE SLOT? 訊息，要求您打入原始磁碟的擴充槽代號。
2.	打入 [6] 表示擴充槽編號爲6，假若磁碟機控制界面卡是插在另一擴充槽，請打入該編號。	螢幕上顯示 DRIVE?，表示要求您打入放有“原始磁碟”的磁碟機代號。
3.	打入	螢幕上顯示DESTINATION SLOT

	[1] 表示是第一部磁碟機，請按下[RETURN]鍵。	?，要求您打入目的磁碟的擴充槽編號。
4.	打入 [6] 表示擴充槽編號爲 6，假若磁碟機控制界面卡是插在另一擴充槽，請打入該編號。	螢幕上顯示 DRIVE? ，表示要求您打入放有“目的磁碟”的磁碟機代號，記住，該磁碟片必須已經啓始化動作。
5.	打入 [2] 表示是第二部磁碟機，假若您有一部磁碟機，請改爲[1]	FILEM 應顯示它用得著的擴充槽及磁碟機代號。到目前爲止尚未複製任何資訊。FILEM 祇是知悉該從何處取得資訊和儲存資訊至何處。 FILENAME? 訊息顯示出來，等待您打入檔案的名稱。
6.	在此時，您可以使用通配字元來表示“一部份”的檔案。假若您想拷貝 MASTER 檔案，您可以打入[M][A][S][T][E][R]或[M][=][R]或者[=][M][=]名稱。但是，[=M=]名稱表示法將會複製檔名含有M的 5 個檔案，它們可能不完全是您希望的檔案。雖然您已經知道用通配字元，如 M=R 可以“抄捷徑”的複製一個或多個檔案，我們仍建議您採用下	

	列的“複查提示功能”來對各個指令行使指令。祇要使用了“通配字元”表示法，程式會顯示DO YOU WANT PROMPTING? 訊息；若您希望有個機會決定是否拷貝某個合乎該表示法的檔案，請打入[Y]。而後，您可以針對各別的檔案，以[Y]表示“已確定”，以[N]表示跳過該檔案。	
	假若您祇希望拷貝一個檔案，最好的方法是打入完整的檔名，如[M][A][S][T][E][R]並按下 [RETURN] 鍵。	顯示的INSERT DISK(S) PRESS <ESC> TO RETURN TO MAIN MENU OR ANY OTHER KEY TO BEGIN訊息表示程式已準備好要執行了。
7.	確定原始磁碟片已放在第一部磁碟機，而已啓始化磁碟片放在第二部磁碟機。若您已準備好要進行拷貝工作，請按下 [SPACE] 鍵。	**FILEM** 程式找出您希望的檔案並立即進行拷貝工作。它會顯示 FILE MASTER. 字樣。假若在第二部磁碟機的磁碟片含有相同檔名的磁碟，則程式會發出“嗶”聲，同時 **FILEM** 會顯示更多的指示。若拷貝工作已完成，則顯示 DONE 字樣。 最後，螢幕上顯示出PRESS ANY KEY TO CONTINUE. 訊息。
8.	按下 [SPACE] 鍵。	**FILEM** 程式功能表畫面重現於螢幕上。

複製已鎖定的檔案 —— FILEM 程式能拷貝已鎖定的檔案，同時在目的磁碟的拷貝版檔案也同時設定為鎖住狀態。

〈2〉 *CATALOG* （目錄）

本選擇項會顯示一磁碟片的目錄。當您在使用 FILEM 程式中有必要去查看一磁碟的內容時，它提供很大的便利：您不必從 FILEM 程式跳脫出來才下達 DOS 的 CATALOG 指令，然後又執行 RUN 指令回至 FILEM 程式。

例如，您想查看 SAMPLE PROGRAMS 磁碟的目錄，自第一部磁碟機中取出 SYSTEM MASTER 磁碟，插入 SAMPLE PROGRAMS 磁碟，而後選擇第 2 項（CATALOG），FILEM 會詢問希望的擴充槽和磁碟機代號；記得，要指明是第一部磁碟機。FILEM 依令行事，顯示 SAMPLE PROGRAMS 磁碟的目錄。按下 [SPACE] 空白鍵後，可回至功能表畫面上。

〈3〉 *SPACE ON DISK* （剩餘的磁碟空間）

當您選擇了第 3 項時，FILEM 程式會回報目前磁碟上已用去的磁區（ sector ）總數以及可用的磁區總數。

請試試看！讓 SAMPLE PROGRAMS 磁碟放在第一部磁碟內，選擇第 3 項，將會顯示類似如下的訊息：

```
SPACE ON DISK

0117 SECTORS FREE
0443 SECTORS USED
```

按下 [SPACE] 鍵，立即回至功能表畫面。

〈4〉 *UNLOCK FILES* （檔案解鎖）

本選擇項的目的是解除檔案的鎖定狀態。下列的例子會告訴您如何用 FILEM 去解除某檔案之鎖定狀態。確定好SAMPLE PROGRAMS 磁碟已放在第一部磁碟機。

操作程序	結　果
1. 自功能表內選取第 4 項並按下 RETURN 鍵。	螢幕顯示 UNLOCK FILES 字樣。FILEM 程式會詢問原始磁碟片所在的擴充槽及磁碟機代號（假若先前 FILEM 並未詢問如此的問題）。
2. 打入對應於檔案的磁碟片所屬的擴充槽及磁碟機號碼。	FILENAME? 顯示於螢幕上。
3. 打入 H E L L O 並按下 RETURN 鍵。您可以在 FILENAME? 之後打入含有通配字元的檔名。若含有通配字元，則會繼續顯示 DO YOU WANT PROMPTING? ，若打入 Y 則表示您希望有個機會再次複查，每回顯示一個檔案，讓您打入 Y 或 N 來選擇是否解除該檔案的鎖定狀態。若您打入 Y 則立即解除指名的	

	檔案鎖定狀態，磁碟機會有動作，接著顯示 DONE 字樣。若打入的是 N，表示不必解鎖，則會在檔案名稱旁顯示 CANCELLED 字樣，接著重複相類似的問句提示	螢幕上顯示 INSERT DISK(S). PRESS <ESC> TO RETURN TO MAIN MENU OR ANY OTHER KEY TO BEGIN. 訊息。
4.	確定希望解除檔案鎖定狀態的磁碟片已放在指定的磁碟機。按下 SPACE 鍵，立即進行解鎖工作。	您將會看到 FILE HELLO 字樣，接著 DONE 顯示出來，表示 FILEM 程式已完成解除指名 HELLO 檔案的鎖定狀態。最後，會顯示 PRESS ANY KEY TO CONTINUE. 訊息。
5.	按下 SPACE 鍵回至 FILEM 功能表畫面。	

〈5〉 *LOCK FILES* （檔案鎖定）

本選擇項將會鎖定指名的檔案，而後此檔案無法被更名、删除或變動內含資訊。

下列是使用 FILEM 鎖定磁碟片內所有檔案的實例，請確定 SAMPLE PROGRAMS 磁碟片已放在第一部磁碟機。

	操作程序	結　果
1.	打入 5 選取功能表內第 5 項，並按下 RETURN 鍵。	LOCK FILES 字樣顯示於螢幕上。FILEM 程式會詢問原始磁碟所在的擴充槽及磁碟機代號（假設先前並未詢問如此的問題），

2.	打入對應的適當號碼，表示那些檔案所在的磁碟片資訊	FILENAME? 顯示於螢幕上，要求您鍵入檔名。
3.	打入 [=] 並按下 [RETURN] 鍵。 一旦您使用了單一的通配字元當作檔名，即表示指定的磁碟內“所有”檔案。	螢幕顯示了 DO YOU WANT PROMPTING? 訊息。
4.	打入 [N] 表示不必複查了，因為您希望鎖定所有檔案。請加按 [RETURN] 鍵。 假若您希望確認每一個需要鎖定的檔案，請打入[Y]，而後您便能對於每一個檔案以[Y]或[N] 加以選擇是否鎖定檔案。	您會看到 INSERT DISK(S) PRESS <ESC> TO RETURN TO MAIN, MENU OR ANY OTHER KEY TO BEGIN. 訊息。
5.	按下 [SPACE] 表示開始動作。	FILEM程式將會重頭檢查合乎您指明的檔名規格，螢幕上會先顯示 FILE 字樣，接著緊跟著處理中的檔案名稱。當 FILEM已已完成所有檔案鎖定狀態的工作，螢幕會出現 DONE 字樣。
6.	按下 [SPACE] 鍵。	螢幕回至 FILEM 功能表畫面。

〈6〉 *DELETE FILES* (刪除檔案)

本選擇項可以刪除一個或一組內數個檔案。**FILEM** 自磁碟內刪除指定的檔案並消去對應於目錄的部份。一旦您刪除了一些檔案，即清出一些空間。

當您選擇了第6項時，DELETE FILES 字樣會出現於螢幕上，並且會向您詢問原始磁碟所在的擴充槽及磁碟機代號（假若先前的操作過程中並未詢問相同的問題）。請查驗在磁碟機內是否放對了磁碟片。

當您看到了 FILENAME? 詢問訊息時，請打入您希望刪除的檔案名稱，並按下 RETURN 鍵。一旦指定的檔案已全部刪除了，FILEM會顯示 DONE 字樣，接著是 PRESS ANY KEY TO CONTINUE 訊息，祇要按下 SPACE 鍵便能回至功能表畫面。

〈7〉 *RESET SLOT & DRIVE* (重定擴充槽及磁碟機代號)

本選擇項的目的是取消先前指定過的擴充槽及磁碟機代號。下一次您選擇到某一個需要擴充槽及磁碟機代號的功能項時，**FILEM** 程式將會向您詢問上述各值。當您希望用 **FILEM** 內的目錄選擇項連續查看兩個不同磁碟機內的磁碟，請選擇本項。

在 **FILEM** 功能表畫面內選擇第7項，螢幕上會顯示 RESET SLOT & DRIVE, DONE, 而後是 PRESS ANY KEY TO CONTINUE. 訊息；按下 SPACE 鍵，可回至功能表畫面。

〈8〉 *VERIFY FILES* （查證檔案內容）

當您選擇了第 8 功能項時，**FILEM** 會查證您指定的檔案內容。亦即是 **FILEM** 嘗試將磁碟內的檔案讀取出來並存入記憶體。**FILEM** 可以查證任何型態的檔案，此選擇項相當於 **DOS** 的 **VERIFY** 指令。

選擇了第 8 功能項後，螢幕上會顯示 VERIFY FILES 字樣，**FILEM** 會向您詢求原始磁碟所在的擴充槽及磁碟機代號（假若 **FILEM** 先前並未問過相同的問題的話）。請確定已放對了磁碟片在磁碟機內。

FILENAME? 提示問句出現時，請打入您希望查證的檔案名稱並後隨一個 RETURN 鍵。一旦檔案已查證無誤，**FILEM** 會顯示 DONE 和 PRESS ANY KEY TO CONTINUE. 訊息，請按下 SPACE 鍵，回至功能表畫面。

假若檔案並不存在於磁碟內，您會得到 FILE NOT FOUND 錯誤訊息。假若某一檔案無法加以查證，則顯示 I/O ERROR 錯誤訊息。

〈9〉 *QUIT* （結束 FILEM 程式）

當您選擇了第 9 功能項時，結束 **FILEM** 並回到 **BASIC**，這時候，螢幕會重現在您執行 **FILEM** 程式之前的 **BASIC** 提示符號。

附註：**FILEM** 程式實際是個可啓動 FID（SYSTEM MASTER 磁碟內的機器碼程式）的 Applesoft **BASIC** 程式。您也可以直截了當的執行 FID 程式，請下達如下的指令：**BRUN FID**。

FILEM顯示的訊息

下列是您使用FILEM程式時，可能會出現於螢幕上的訊息：

- 當您選擇了“複製”功能項，而目的磁碟已含有您指定檔名的檔案時，FILEM會顯示如下的訊息：

```
FILE [filename]
ALREADY EXISTS.
TYPE IN NEW FILENAME FOR THE COPY OR
<RETURN> TO REPLACE EXISTING FILE OR
<CONTROL-C><RETURN> TO CANCEL COPY
:
```

假若您按下 RETURN 鍵，而且目的檔案已解除鎖定狀態，FILEM會將原始檔案的內容覆蓋在目的檔案之上。假若您鍵入的是個新名稱，則原始檔案的內容將以新檔名存於目的磁碟上。

- 當您選擇了“複製”功能項，而且目的檔案已被鎖定，則顯示如下的訊息：

```
FILE LOCKED
DO YOU WISH TO REPLACE IT ANYWAY?
```

假若您打入 Y 並按下 RETURN 鍵即表示“是”（取代原有的）。FILEM不必顧慮這種情況，仍照常將原始檔案的內容覆蓋在目的檔案之上。若打入的是 N，則FILEM給予您一次機會，重新設定新的檔名，此時螢幕上出現如下的訊息：

```
TYPE IN NEW FILENAME FOR THE COPY OR
<RETURN> TO REPLACE EXISTING FILE OR
```

```
<CONTROL-C><>RETURN> TO CANCEL COPY
:
```

若您按下RETURN鍵，表示告訴FILEM直接覆蓋在已鎖定的目的檔案。假若您按下的是CONTROL-C和RETURN鍵則表示取消拷貝動作，FILEM不進行複製功能。假如您僅是複製單一檔案，則顯示CANCELLED字樣並回至功能表畫面。如果您是引用“通配字元”來複製一組（多個）檔案，則僅取消對目前操作中的檔案之拷貝動作。

- 假若FILEM無法找尋到指名的原始檔案，則顯示如下的訊息。

  ```
  NO FILES SELECTED
  ```

  請查驗磁碟的目錄，再度嘗試一次該指令。
- 假若您指定的擴充槽及磁碟機代號是不正確的，或者是錯誤的檔案名稱，則分別顯示如下的訊息。
  INVALID SLOT, INVALID DRIVE, 或 INVALID FILENAME. 。

第4章摘要

指 令

RENAME fn1,fn2 [,Sn] [,Dn] [,Vn]	更改 fn1的檔名為新檔名 fn2。
LOCK fn [,Sn] [,Dn] [,Vn]	預防指名的 fn 檔案無意中被毀損。已鎖定的檔案無法加以更名、刪

	除或者變動其內含資訊，除非您解除檔案鎖定狀態。
UNLOCK fn [,Sn] [,Dn] [,Vn]	解除一已鎖定保護的檔案，讓該檔案往後可以加以更名、刪除或者變動其內含資訊。
DELETE fn [,Sn] [,Dn] [,Vn]	自磁碟中移走（刪除）一檔案。指名 fn 的檔案必須是已解除鎖定保護。
VERIFY fn [,Sn] [,Dn] [,Vn]	查證檔案是否已正確地存放於磁碟上。

程　式

FILEM	將數個常用到的 DOS 指令合聚在一個程式內，讓使用者能很方便地一次同時針對一個或多個檔案進行各種DOS指令操作。FI-LEM程式中可引用“通配字元”於檔名中，而且對於已鎖定的檔案仍然可有效地操作。

5 使用已存於磁碟上的各種程式

■ RUN指令 125
- 實 例 128

■ LOAD指令 129
- 實 例 130

■ SAVE指令 130
- 實 例 132

■ 與其他週邊裝置溝通 133
- PR# 指令 135
  - △實 例 136
- IN# 指令 137
  - △實 例 138

■ 重新啓動DOS的途徑 138

■ 結 論 139

■ 第5章摘要 140

第5章

使用已存於磁碟上的各種程式

本章的目的是介紹一些讓您可使用已存於磁碟上各種 BASIC 程式的 DOS 指令，請您特別留意 RUN 指令，它可以將存於磁碟的 BASIC 程式檔案載入記憶體內，並立即執行 BASIC 程式。

在本章介紹的指令可以讓您

- 將一程式帶入記憶體內並且立即執行它（RUN指令）
- 將一程式帶入記憶體內，但不執行它（LOAD指令）
- 將目前記憶體內的程式或者資訊存入磁碟內（SAVE指令）
- 與連接到主機上的各種週邊裝置"通訊"（資訊交流）

RUN指令

RUN 指令可用來執行已存於磁碟內的Applesoft 或 Integer BASIC 程式。本指令將程式載入記憶體內。讀取檔案內容、載入及啓動一程式的連續過程，常稱之"跑"程式（running the program，直譯爲"執行"一程式）。

RUN 指令包含了數件工作：當 DOS 接到 RUN 指令，首先它會去找尋指定的檔案，接著是將程式內容讀取出來存入記憶體，

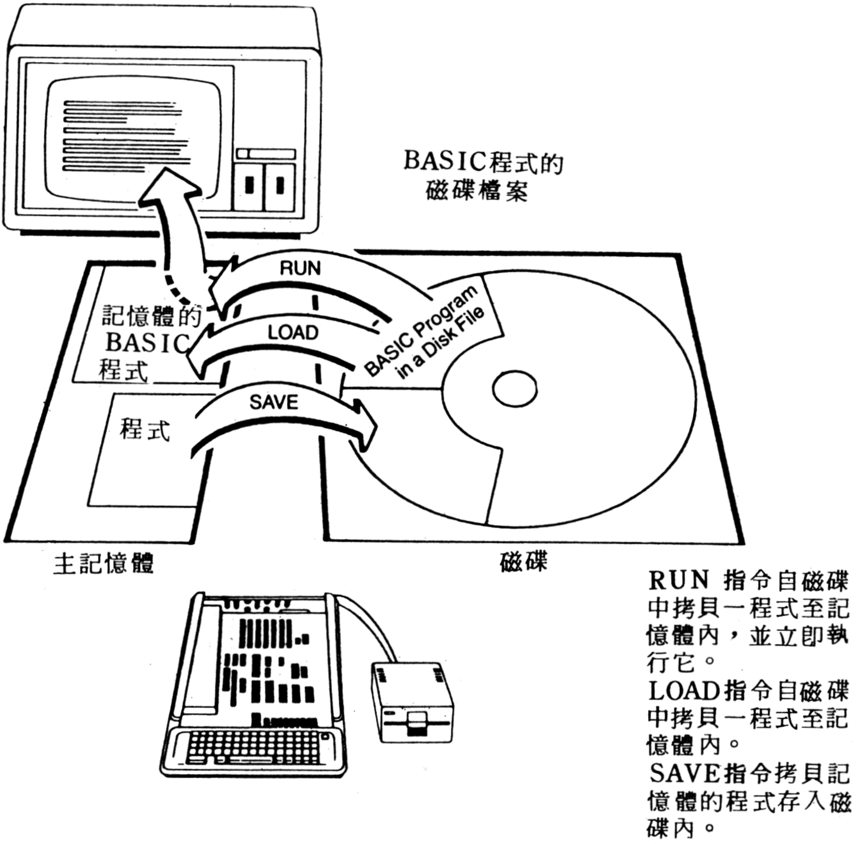


圖 5-1 BASIC程式與DOS指令

最後查驗它的檔案型態，而後嘗試轉換至它所屬的語言系統（假若與目前的語言互不相容，例如目前是在Applesoft BASIC 控制之下，而載入的程式是 Integer BASIC（型態 I））。

當一檔案的型態是A，而 Applesoft BASIC 並未待命中（

active)，則 DOS 轉換至Applesoft 控制之下。同理，若一檔案型態是I，而Integer BASIC 並未待命中，則 DOS 負責轉換至Integer 系統控制之下。

注意，假若您的Apple電腦僅能使用一種 BASIC 語言，則 DOS 無法自動地轉換至適用的語言系統。Apple II 或Apple II Plus 機種，需要附加一語言界面卡(或16K RAM 記憶體界面卡)或者能使用Applesoft 和 Integer BASIC兩種語言的其他週邊裝置界面卡。

一旦 DOS 已經轉換完成適用的程式語言， DOS 會讀取磁碟內指定的程式，將其內含資訊存入記憶體內並立即啓動它。既然 RUN 指令自動地載入程式至記憶體內，您就不必在執行一程式之前下達LOAD指令。(關於LOAD指令請見下一節的說明。)

順便提醒您， RUN 指令會在載入程式之前，先行清除主記憶體內的任何東西。假若您不希望已存於記憶體的資訊漏失掉，請在下達 RUN 指令之前，利用SAVE指令將它們儲存於磁碟上。

RUN指令的語法如下：

```
RUN fn [,Sn] [,Dn] [,Vn]
```

指令中的 fn 表示您希望由 DOS 執行的檔案名稱，該檔案的型態必須是A或是 I。假若指定的檔案並未存在於磁碟上，則出現 FILE NOT FOUND.錯誤訊息。假若其檔案型態不是A也不是I，則顯示 FILE TYPE MISMATCH. 錯誤訊息。至於擴充槽、磁碟機及卷號等引數均是可有可無的選擇項，您祇需在不想沿用目前的各引數預設值情況下，才有必要加上這些引數。各引數的意義，請見第2章說明。

順便一提：祇要程式已存在於記憶體，您可以省略 RUN 指令中的檔案名稱，直接下達 RUN 指令。若不附加檔案名稱此引數， RUN 即是一 BASIC 指令。

實 例

下列例子是展示 RUN 指令如何與 Integer BASIC 和 Applesoft BASIC 配合使用。

（假設您僅有一種 BASIC 語言*：如果您嘗試執行不配備某一程式所屬的語言系統，則會出現 LANGUAGE NOT AVAILABLE. 的錯誤訊息，請用 RUN 指令執行符合您的BASIC 型態的程式。）

1. 將 SAMPLE PROGRAMS 磁碟片插入第一部磁碟機。利用 CATALOG 指令查看該磁碟的目錄以及它所在的擴充槽及磁碟機代號。打入

   [C][A][T][A][L][O][G][SPACE][D][1][,][S][6]

   您將在目錄中查到 APPLEVISION 和 COLOR TEST 程式名稱。

2. 現在，嘗試執行 Integer BASIC 型態的程式。打入

   [R][U][N][SPACE][A][P][P][L][E][V][I][S][I][O][N]

   並按下 [RETURN] 鍵，您將會看到相當奇妙的畫面。若您想中止畫面的展示，請按下 [ESC] 鍵。此時，Integer BASIC 的提示符號>出現於螢幕上。

3. Applesoft BASIC 型態的 COLOR TEST 程式是專門設計來測試彩色顯示器的特性，假若您並無彩色的顯示畫面，則出現的是兩種對比顏色（ two-tone ）的畫面。打入

   [R][U][N][SPACE][C][O][L][O][R][SPACE][T][E][S][T]

任何時候您可以按下 RETURN 鍵中止程式，並回至功能表畫面。請按下 5 以結束 COLOR TEST 的程式，此時，Applesoft 的提示符號] 出現於螢幕上。

語言系統的轉換：假若您的 Apple 電腦同時配備有 Applesoft 和 Integer BASIC 兩套語言系統，DOS 會自動切換成適用的語言，您並不需下達特別的指令。

LOAD指令

LOAD指令的作用是將磁碟上的程式內含資訊轉存至記憶體。在將程式載入記憶體之前，DOS 先行查驗程式的檔案型態，而後設法切換至適用的 BASIC 語言系統上。當您利用LOAD指令將一程式的內容拷貝至記憶體時，存放於磁碟的檔案內容並未受到破壞。

在您使用 RUN 指令之前，並不需將一程式載入記憶體，因爲 RUN 指令會自動將程式載入記憶體內。

提醒您，在LOAD指令試著將一程式內容存入記憶體之前，DOS 會先行清除記憶體，所以假若您不想漏失任何既存於記憶體的資訊，請在下達LOAD指令之前先用SAVE指令儲存起來。

本指令的語法如下：

```
LOAD fn [,Sn] [,Dn] [,Vn]
```

指令中的 fn 表示您想載入的程式之檔案名稱，它的檔案型態必須是A或 I 。假若指定的程式並不存在於磁碟上，則顯示FILE NOT FOUND. 錯誤訊息，假若其檔案型態不是A也不是 I ，則顯示的是 FILE TYPE MISMATCH. 錯誤訊息。至於擴充槽，磁碟機代號及卷號等引數均是可有可無的選擇項。只在您不想沿用目前各引數的

預設值情況下，您才有必要選用上述各引數。各引數的意義詳見第 2 章說明。

實 例

1. 將SAMPLE PROGRAMS 磁碟片放在第一部磁碟機，把其中的GET TEXT程式之內容拷貝一份存入記憶體內，請打入

   L O A D SPACE G E T SPACE T E X T

   您會察覺到磁碟機在忙碌著，而後螢幕重現 BASIC 提示符號。

2. 查看該程式的內容，請使用 LIST 指令。打入

   L I S T

   您會看到構成GET TEXT程式內容的 BASIC 指述（statements）。假若您對這程式的動作原理有興趣的話，請參考"DOS 程式設計者手册"。

（附註：當您載入一程式時，DOS 祇是將磁碟上的程式拷貝其內容送入記憶體內，存於磁碟上的程式仍保持不變。您可以利用 CATALOG 指令再度確認一下，其目錄仍含有該程式的記錄項。）

SAVE指令

SAVE指令的作用是將目前已存於記憶體內的 BASIC 程式，以檔案的形式儲存於磁碟上。

DOS 已記錄著您目前使用中的語言型態，它會將該程式以此語言型態當成程式的檔案型態。當您在使用 Applesoft BASIC ，DOS 即以檔案型態 A 儲存該程式；假若是使用 Integer BASIC，則以檔案型態 I 儲存該程式。DOS 在儲存程式之時，已記錄該程式的實際長度和該程式儲放於磁碟的位置。

一旦您已儲存某個程式，可以接著下達LOCK指令為該程式加上鎖定保護以防止無意中的刪除行為。UNLOCK 指令可以解除程式的保護狀態。

您可以利用SAVE指令完成“複製檔案”的工作，產生一程式的多份拷貝，方法如下：利用LOAD指令將一檔案內容存入記憶體內，而後以SAVE指令（附加原來的檔案名稱）儲存該程式至不同的磁碟片上，或者改用不相同的檔案，將該程式儲存於相同的磁碟片上。

本指令的語法如下：

```
SAVE fn [,Sn] [,Dn] [,Vn]
```

fn 項引數是代表該程式的檔案名稱，假若某一檔案已以相同的檔名存於磁碟上，那麼在該重名檔案不是鎖定狀態，而且也具有相同的檔案型態（A或I）的兩個前提下，DOS 會將記憶體內的程式覆蓋寫入已存於磁碟上的程式。

假若在使用SAVE指令時，指定的磁碟片已壞了，或者磁碟片尚未啓始化，或是磁碟機內並無磁碟片，或是磁碟機的門沒關好，那麼您會得到 I/O ERROR 的錯誤訊息。請依據實際的情況，將已啓始化的磁碟片放入磁碟機內，關好磁碟機的門，而後再下達一次SAVE指令即可。

語法中表示的擴充槽、磁碟機代號及卷號等引數均是可有可無的選擇項；您只需在不想沿用目前各引數之預設值情況下，才有必要加選上述各引數。引數的意義請參考第 2 章說明。

注意，假若您儲存記憶體內的程式至一磁碟上時，已有一相同檔名及型態的檔案存在於磁碟，那麼在磁碟上的程式內容將會因新程式的覆蓋寫入而漏失所有內容。DOS 並不會顯示任何警告的或表示操作錯誤的訊息。

實 例

1. 將一片已啓始化磁碟放在第一部磁碟機上。請不要加上寫入保護貼紙——亦即是“可寫缺口”不要覆蓋任何貼紙。
2. 下達NEW 指令為一新程式預先清除記憶體的內容（NEW是 BASIC 的指令），打入

   NEW

   並按 RETURN 鍵。
3. 現在，請打入一簡短的程式，例如：

```
10 PRINT "A SHORT PROGRAM"
20 END
```

   即使您並不十分熟悉程式設計的要領，您也能猜到這個程式只作一件事情：顯示 A SHORT PROGRAM 訊息。
4. 測試本程式，請使用 BASIC 中的 RUN 指令。打入

   RUN

   並按下 RETURN 鍵。
5. 已測試無誤後，可以用SAVE指令將此程式以您希望的檔案名稱儲存至磁碟上。例如，打入

   SAVE SPACE ONE SPACE LINER

磁碟機開始忙碌著，將命名為 ONE LINER 的程式儲放至磁碟上。

在您使用SAVE指令之後，您可以接著下達 CATALOG 指令查看該程式已在磁碟上佔有一席之地。請注意到該程式的內容仍保留在記憶體上（除非您又下達了 NEW 指令，或者關掉電腦），您只要使用 RUN 或 LIST 指令就可查驗該程式確實仍在記憶體內。

與其他週邊裝置溝通

PR # 和IN # 指令可以讓您的Apple 電腦與其他連接在擴充槽的週邊裝置（例如印表機）互相溝通資訊。

您的Apple 電腦經常是將文字送至螢幕上，並從鍵盤上取得您打入的文字。 PR #和 IN # 指令的目的是另闢一條溝通管道，讓Apple 電腦可以與其他裝置“對話”（溝通資訊）。這些其他裝置，常稱之為"**週邊裝置**"（peripheral device）至於“溝通資訊”即表示“進行 I/O 動作”，亦即是“輸入和輸出資訊”。

注意，這兩個指令中的#字元不要遺漏了，請完整的鍵入PR#或IN# 指令名稱。

警告：若在 PR # 或 IN # 指令中指定的擴充槽中並無**控制界面卡**，可能會有意想不到的事故。例如，若一控制界面卡插在編號 7 的擴充槽內，而您下達了PR #6指令，可能會啓動其他系統，但您不會得到錯誤訊息的回報，可能是 DOS 已經認為Apple 主機與某裝置已斷絕通訊。若 DOS 已無法自己啓動系統，您必須啓動 DOS 。請參考本章後頭的“重新啓動 DOS 的途徑”說明。

以 APPLESOFT 為檔名，保存此檔，並設定鎖住狀態。更改HELLO和APPLESOFT，使它可執行您的轉鑰系統，而非原先執行中的 BASIC 系統。

7. 將您的轉鑰程式取名爲TURNKEY。以下的例子是用 Applesoft 所寫的TURNKEY程式：

```
100 TEXT : HOME
200 VTAB 3: PRINT "THIS IS MY TURNKEY
      PROGRAM "
300 VTAB 5: FOR A = 1 TO 10: PRINT A: NEXT A
400 PRINT : PRINT "THIS IS THE END OF MY
      TURNKEY PROGRAM"
500 END
```

8. 確定您的磁碟有HELLO、APPLESOFT、INTBASIC、FPBASIC、LOADER. OBJ0和TURNKEY檔案，檢驗您的轉鑰式磁碟，將它放入第一部磁碟機，關掉電源並重開。磁碟重新啓動DOS，若有需要則載入 Integer BASIC 或Applesoft ，然後自動執行轉鑰程式TURNKEY。

5.7 摘 要

5.7.1 指 令

EXEC fn [, Rn] [, Sn] [, Dn] [, Vn]

指示 DOS 從指令檔（command file）讀取指令，而不是由鍵盤。fn引數指定一指令檔，該檔係由 BASIC 程式所建立。指令檔可包含 DOS 指令、資料和 BASIC 指述。

MAXFILES n

其中n是由1至 16的任一整數，指定在任一瞬間可同時開啓

PR＃ 指令

PR ＃ 指令的目的是告知您的Apple 電腦傳送資訊至您指定的裝置上，亦即是連接於指定擴充槽的裝置。 PR ＃ 指令並傳送“ 所有 ”的資訊——包括提示符號、錯誤訊息、指令、程式內容列表和目錄，至指定的裝置。（ 註：關於 PR ＃ 此名稱有點歷史背景，值得您注意。在電腦早期的發展過程中，印表機（ printer ）是當時唯一的一種輸出裝置，也許 PR ＃即取意於“ printer ”。印表機須賦予一個編號，而＃字元總常用來代表“ 數目 ”，即以“PR ＃”此名稱形成一個奇怪而有趣的指令名稱。）

本指令的格式如下：

PR＃n

指令中的數目，n ，是代表資訊流程中“ 接收資訊 ”裝置所在的擴充槽編號。此數目祇能是0 至7 之間的任一數。

當您希望回復成資訊送至顯示螢幕的情況，請使用 PR＃ 指令（ 若是使用80 格文字顯示界面卡則 PR＃0 無效 ）。在Apple IIe機種的主機內雖無編號0 的擴充槽，但PR＃0 仍然能正常作用。

注意：若Apple IIe 機種電腦是配合著80 格文字顯示界面卡（ 80-columntext card ）使用，我們並不建議您使用 PR＃0指令。請改用 PR＃3 指令讓印表機不再作用，同時又重建80 格文字顯示界面卡的溝通管道，接著再按下 [ESC] [CONTROL]－[Q] 令印表機及 80 格顯示卡不再有作用。

假若您有部印表機，您可以傳送磁碟的目錄至印表機上。假設您已將印表機控制界面卡插在編號1 的擴充槽，請打入

[P][R][SHIFT][#][1][RETURN]

[C][A][T][A][L][O][G][RETURN]

當已經傳送至印表機，希望回復資訊送至螢幕的情況，打入

P R SHIFT - # 0

或

P R SHIFT - # 3　（假若接上了 80 格顯示卡）

利用 PR # 指令啓動一磁碟：雖然 PR # 指令的原意是用來指明資訊傳送的"接收端"裝置，但是您可以用 PR # 指令來啓動磁碟機。利用此方法可以比關掉電腦再重開電源的方法來得簡單省時。（註：PR # 6 是重啓磁碟的捷徑）。

注意：若您的 Apple II 電腦是配備著舊版的"監督程式ROM"，您須按下 6 CONTROL - P 以啓動 DOS，並在使用 PR # 指令之前已回復成 BASIC 語言狀態。

實　例

1. 將 SAMPLE PROGRAMS 磁碟放在第一部磁碟機，關好磁碟機，打入

   P R SHIFT - # 6

   並按下 RETURN 鍵。若您的磁碟機控制卡是連接在另一個擴充槽上，請改成對應的擴充槽編號。第一部磁碟機會開始運轉，亦即是 DOS 啓動了磁碟機。

2. 假若已將印表機界面卡插在編號 1 的擴充槽，現在，想將程式內容自印表機列印出來。首先，將 HELLO 程式讀取並存入記憶體，打入

   L O A D SPACE H E L L O

3. 改變電腦的資訊傳送對象爲印表機，打入

P R SHIFT # 1

並按下 RETURN 鍵。假若印表機界面卡是插在另一擴充槽，請改成對應的編號。

4. 現在，您可以列印程式的內容，打入

L I S T

並按下 RETURN 鍵，您的印表機即開始印出HELLO程式的內容。

5. 嘗試下達其他的指令。平常是顯示於螢幕的各種東西，現在改在印表機上的報表機列印出來，包含指令本身，提示符號以及發生的錯誤訊息。

6. 回復資訊傳送至螢幕的正常情況，請使用 PR # 指令，必須指明是編號 0 。

IN # 指令

IN # 指令是告訴您的電腦自何處取得資訊，亦即是指明連接在某一擴充槽的某個裝置。 IN 即代表“ input ”（輸入）。

本指令的格式如下：

```
IN#n
```

此處的 n 是代表資訊讀取（輸入）裝置所在的擴充槽編號。 n 是 0 到 7 之間的整數。當您使用了 IN # 指令， DOS 就會一直自指定的擴充槽之裝置取得（讀取）它希望的資訊。

若您下達 IN # 0 指令，表示回復自鍵盤讀入資訊的正常情況

。本指令同樣適用於Apple IIe 機種，雖然在 Apple IIe 電腦並無編號 0 的擴充槽。

實 例

1. 假若您的Apple電腦內，編號 4 的擴充槽插上了電話數據傳輸界面卡（ modem ），此界面卡連上了電話網路（ telephone network ），那麼您可以下達如下的指令，讓 DOS 自網路中讀取資訊。

   I N SHIFT-# 4

2. 回復讀取鍵盤輸入的狀況，使用 IN# 指令，指明是編號 0 的擴充槽之裝置。

註：您可以使用 IN # 指令啓動第一部磁碟機的磁碟——您須指明插有磁碟機控制界面卡的擴充槽編號，就如同您使用 PR # 指令的情況。

重新啓動 DOS的途徑

在下列的情況出現時，您可能需要重新啓動DOS：鍵盤輸入任何字元均無作用，或游標消失不見（人們常稱為電腦系統“當掉”了）。您可能會想用一新磁碟來重新啓動DOS。

當您曾用 DOS 啓動電腦系統後，有幾種方法可以重新啓動DOS。下列方法是依其繁煩性及影響系統的程度排列的：

1. 打入 CONTROL-RESET 。本法不會影響到主記憶體內含資訊。假若系統已當掉了，這是第一個您應該

嘗試的方法。

2. 打入 C A L L SPACE 1 0 0 2 。這是一個 BASIC 指述，可以重新連接上 DOS ，但不會影響到主記憶體的內容。

3. 打入 P R SHIFT-# 6 重新啓動 DOS ，實際上它是再啓動所有 Apple 機種電腦內接在"編號 6 的擴充槽"的"第一部磁碟機"。或者在 Apple IIe 機種電腦按下 -CONTROL-RESET 啓動系統。這兩種方法均會破壞既存於記憶體的內容。在測試一新的系統磁碟能否啓動，您應該採用本法。

假若螢幕上顯示監督程式的星號提示符號(*)，請打入

6 CONTROL-P　　或者

6 CONTROL-K　　重新啓動 DOS。

您也可以打入

3 D 0 G　（注意，它內含的是零 0 不是字母 O ）

或

C 6 0 0 G（本式亦含有零）重新啓動 DOS。

假若您的磁碟機控制界面卡是插在編號 6 以外的另一擴充槽，則在上述指令改爲對應的編號代入 6 位置。

結　論

到此爲止，DOS 的簡介已告一段落，您已經知道不少複雜的磁碟操作動作及其指令。如果您已完全了解這些基本的指令及操作概念，您可以進一步地閱讀" DOS 程式設計者手册 "的詳細章節。

第5章摘要

RUN fn [,Sn] [,Dn] [,Vn]	自磁碟上拷貝一程式（其檔案型態是A或I）存入記憶體內，並立即執行此程式。
LOAD fn [,Sn] [,Dn] [,Vn]	自磁碟上拷貝一程式（其檔案型態是A或I）存入記憶體內。一旦程式已存於記憶體內，您可以執行它，修改它，或者再儲存於磁碟上。
SAVE fn [,Sn] [,Dn] [,Vn]	將目前存於記憶體的 BASIC 程式，拷貝一份儲存於磁碟上。若是Applesoft 程式，則其檔案型態爲A；若是 Integer BASIC 程式，則其檔案型態爲I。
PR# n	指明您的電腦傳送資訊至何處，亦即是將輸出送至連接某一擴充槽上的裝置。
IN# n	指明您的電腦應自何處讀取資訊，亦即是輸入資訊的某一擴充槽上的裝置。

附　　錄

附錄A：轉換13磁區磁碟的格式　145

■轉換13磁區的磁碟：CONVERT13程式　146
- 實　例　147
- 重複名稱的檔名　151
- 通配字元　151

■不需轉換即可執行13磁區磁碟的程式　152
- 使用START 13程式　152
  △實　例　153
- 使用BASICS磁碟　154

附錄B：DOS錯誤訊息　157

■DOS錯誤訊息　157
- DISK FULL　158
- END OF DATA　159
- FILE LOCKED　159
- FILE NOT FOUND　159
- FILE TYPE MISMATCH　160
- I/O ERROR　161
- LANGUAGE NOT AVAILABLE　162
- NO BUFFERS AVAILABLE　163
- NOT DIRECT COMMAND　164
- PROGRAM TOO LARGE　164

• RANGE ERROR 165
• SYNTAX ERROR 166
• VOLUME MISMATCH 166
• WRITE PROTECTED 166
■停止程式的執行 167

附錄C：DOS磁碟內的各種程式 169

■SYSTEM MASTER磁碟內的程式 169
■SAMPLE PROGRAMS磁碟內的程式 171

附錄D：DOS操作概念及指令摘要 175

■DOS操作概念 175
• 啓 動 175
• 重新啓動DOS 176
• 磁碟容量 177
■記號說明 177
• 語 法 178
• 引 數 178
■指令摘要 179
• CATALOG 180
• DELETE 181
• IN# 181
• INIT 181
• LOAD 182
• LOCK 183
• PR# 183
• RENAME 184
• RUN 184

- SAVE 184
- UNLOCK 185
- VERIFY 185

附錄A

轉換13磁區磁碟的格式

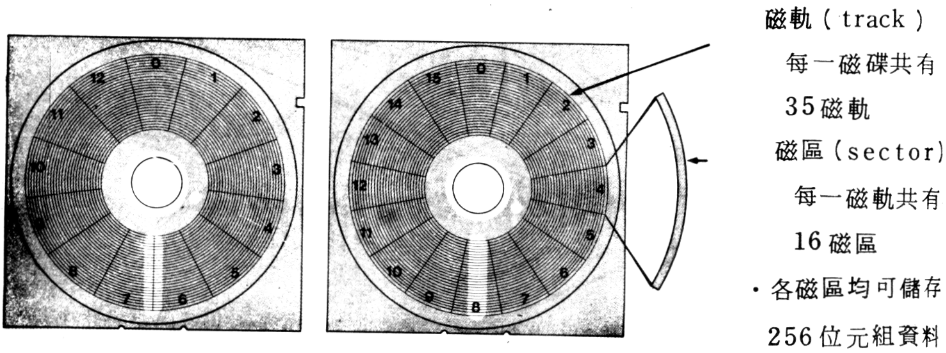
本附錄的適用對象是那些手上有遊戲程式或應用程式的磁碟，卻無法在目前的電腦系統正常運轉的使用者。若您啓動一磁碟，而磁碟機卻在不停地空轉，您也檢查過磁碟片已擺放妥當，那麼也許磁碟片是屬於 13 磁區的格式。

早期版本（ 3.1 和 3.2 ）的DOS是將磁碟的表面規劃成每一磁軌（ track ），含有 13 個磁區（ sector ），目前您擁有的DOS（版本 3.3 ）則規劃成 16 個磁區（ 圖A-1）。13磁區到 16磁區的轉換可以令每一磁碟的可用空間增加約 20 ％。

但是，13磁區格式的磁碟片無法與 16 磁區的DOS程式配合使用，除非它們經過一次格式的轉換。本附錄的目的即是告訴您：

- 利用 CONVERT13 程式轉換 13 磁區格式之磁碟
- START13 程式讓您目前系統可使用 13磁區格式之磁碟
- 使用 BASICS 磁碟，假若因磁碟已有“拷貝保護”無法轉換 13 磁區磁碟時

（附註：一**磁區**（ sector ）是一磁軌中的一固定容量的片斷部份。DOS 3.3 在啓始化一磁碟時，是將磁碟表面規劃成 35 個磁軌，每一磁軌再細分 16 磁區，以備往後存取資料之用。每一磁區可儲存 256 位元組（ byte ）的資料。）



13磁區格式　　16磁區格式

圖A-1　磁碟：13磁區與16磁區格式的比較

轉換13磁區的磁碟：CONVERT13程式

SYSTEM MASTER磁碟內的 CONVERT13 程式其作用是將一13磁區的磁碟（包括所含的程式）轉換成16磁區格式的磁碟。CONVERT13 是將13磁區磁碟內容全部複製至另一已啓始化的16磁區格式之磁碟上。CONVERT13 祇是讀取13磁區磁碟的內容，而後寫入16磁區磁碟，並未破壞任何存在於13磁區磁碟的資訊。

附帶一提，您可能較習慣用MUFFIN 程式來轉換早期版本的DOS，實際上，MUFFIN 程式執行相同於CONVERT13 的功能，因為 CONVERT13 程式啓動MUFFIN 的執行。

實　例

下例將說明用 CONVERT13 程式轉換 13 磁區磁碟的內含資訊至 16 磁區的磁碟上。注意，16磁區磁碟必須已經作過“啓始化”動作。（關於啓始化的步驟，可參考第 1 章的“預備一空白磁碟：INIT 指令”。）

假若您的 13 磁區磁碟已賦予“拷貝保護”（即該磁碟經過特殊處理後無法複製該磁碟），您無法進行轉換的工作，但您仍然可藉助於 START13 程式和 BASICS 磁碟來執行該磁碟上各程式。（參考本附錄的“不需轉換即可執行 13 磁區磁碟的程式”一節）

本例是在一部磁碟機情況下，示範 CONVERT13 程式的操作程序。您在轉換整片磁碟的過程中，可能須在中途抽換磁碟數次。

兩部磁碟機：若您擁有兩部磁碟機，請在轉換之前，先將 13 磁區的磁碟放在第一部磁碟機，而新準備好的“目的”磁碟（已作過 16 磁區的啓始化工作）放在第二部。當 CONVERT13 程式向您詢問擴充槽及磁碟機代號時，兩者的擴充槽部份打入 6（假設編號 6 是磁碟機控制界面卡所在的擴充槽），隨後分別打入 1 和 2 表示原始（ source ）和目的（ destination ）磁碟機代號。

操作程序	結　果
1. 將 SYSTEM MASTER 磁碟放在第一部磁碟機，打入 [R][U][N][SPACE] —— [C][O][N][V][E][R][T][1][3] 並按下 [RETURN] 鍵。	螢幕上顯示 EXECUTING MUFFIN 簡短訊息，隨後，出現如下圖的 CONVERT13 的功能表

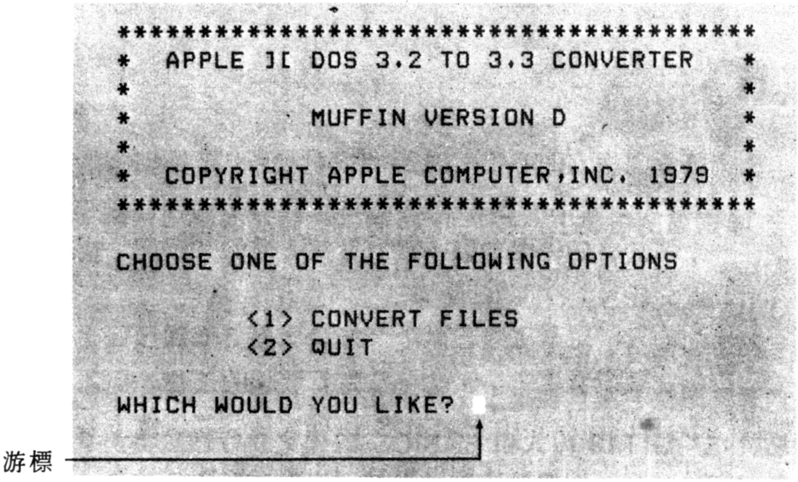


圖 A-2　MUFFIN 標題畫面

2.	進行轉換工作，打入 [1] 並按下 [RETURN] 鍵	程式被啓動後，您會看到 SOURCE SLOT? 問句，表示請您打入(原始)13 磁區磁碟所在的擴充槽編號。隨後，再打入磁碟機代號
3.	告訴程式，磁碟機是接在編號 6 擴充槽的界面卡，打入 [6] 並按下 [RETURN] 鍵。 若您的磁碟機控制界面卡是插在不同的擴充槽，請打入對應的編號	下一個顯示於螢幕的問句爲 DRIVE?
4.	指明放有 13 磁區磁碟片的磁碟機代號，打入 [1] 並按下 [RETURN] 鍵，	接著，再顯示一列問句．DESTINATION SLOT? ，表示請您打入轉換的目的磁碟所在之擴充槽編號。目的磁碟必須是16

	表示是第一部磁碟機。	磁區格式的磁碼。
5.	打入 [6] 並按 [RETURN] 鍵表示編號 6 的擴充槽。若您的磁碟機控制界面卡是插在另一擴充槽，則打入對應的編號	螢幕顯示 DRIVE?
6.	打入 [1] 並按 [RETURN] 鍵，表示第一部磁碟機。到目前爲止，您已經告訴CONVERT 13 原始和目的磁碟均使用同一部磁碟機	現在，程式詢問檔案的名稱，螢幕顯示：FILENAME?
7.	打入 [=] 並按下 [RETURN] 鍵。此等號是代表磁碟內所有檔案的特殊字元。	螢幕顯示 DO YOU WANT PROMPTING? 問句，其作用是給予您再一次的機會，是否需要在 CONVERT13 開始轉換之前，複查一下各個檔案。
8.	打入 [N] 並按下 [RETURN] 鍵，表示“NO”，不需要複查。所有檔案均自動進行轉換。	螢幕顯示一段訊息：INSERT DISK(S), THEN PRESS <ESC> TO RETURN TO MAIN MENU OR ANY OTHER KEY TO BEGIN. 此段訊息再給您反悔的機會，假若不願進行轉換，則此時可按下 [ESC] 鍵，可取消此項工作。

（**註**：假若您並不想轉換 13 磁區的磁碟內所有檔案，則改按下[Y]，而後，程式會一一顯示各個檔案名稱，您可以分別打入[Y]或

N表示“要”或“不要”轉換該檔案；或者打入Q表示結束轉換工作，直接返回功能表。)

9.	自第一部磁碟機取走 **SYSTEM MASTER** 磁碟	本步驟之後，程式工作於兩種磁碟中—13磁區格式和您準備好的16磁區格式之磁碟。
10.	將13磁區的磁碟放入磁碟機，按下 RETURN 鍵開始轉換工作。	**CONVERT13** 找尋在13磁區(原始)磁碟的第一個檔案，並顯示其檔名： FILE HELLO 。接著顯示
11.	將已經下達 **INIT** 指令，作過啓始化的16磁區磁碟放到磁碟機內，再按下 RETURN 鍵。	**CONVERT13** 轉換檔案，將它存放於此新磁碟，並回報 DONE 字樣。
12.	假若在原始磁碟尚有多個檔案，則重複第10、11步驟。**CONVERT13** 會一直指示您再放入原始磁碟於磁碟機。	
	程式重複執行第10和第11步驟，自原始磁碟讀取檔案，轉換，寫入檔案至目的磁碟，直到各個13磁區格式的檔案全部轉換完成並存到新磁碟上。	最後，您會看到 PRESS ANY KEY TO CONTINUE. 的訊息，表示 **CONVERT13** 已經完成轉換所有檔案的工作。
13.	按下 RETURN 鍵。	螢幕重現 **CONVERT13** 的功能表畫面。
14.	回至 **BASIC** 提示符號，打	

入[2]
並按下[RETURN]，表
示"離開"（Quit）此程
式。

現在，您手上已有轉換後的 16 磁區格式的磁碟，它們適用於 **DOS** 3.3 系統，請參考 **DOS** 3.3 手册。

重複名稱的檔名

假若您想轉換的 13 磁區之檔案在目的磁碟有相同檔名的檔案，則您會看到：

```
FILE [filename]
ALREADY EXISTS.
TYPE IN A NEW FILE NAME FOR THE COPY
OR <RETURN> TO REPLACE EXISTING FILE
OR <CONTROL-C><RETURN> TO CANCEL COPY
:
```

您可以採行下列三種措施之一：打入轉換後的新檔案名稱；直接按[RETURN]鍵表示以目前的 13 磁區格式之檔案取代在 16 磁區磁碟的同名檔案；或按[CONTROL]-[C]和[RETURN]中止轉換工作。

通配字元

您在 CONVERT13 的操作中已經見識到"通配字元"（Wildcard Character）的用途：利用一個等號（＝）來代表磁碟內所有檔案。您也可以在檔名中使用此等號，代表任何字元組成或某種類的一組檔名，例如，您可以對 FILENAME? 詢問訊息打入[F][=][L]

[E]，指示CONVERT13將所有以F開頭且LE結尾命名的檔案進行轉換工作。

當您使用了通配字元，CONVERT13 會詢問DO YOU WANT PROMPTING?。假若您希望 CONVERT13 在找到一個13磁區格式的檔案，在轉換之前，讓您複查並確認是否合乎您希望的檔案，則打入[Y]並按[RETURN]鍵。如果您是希望轉換以"通配字元"指定的一組檔名內所有檔案，不需作複查確認的工作，則打入[N]並按[RETURN]鍵。在過程中，可打入[Q]，並按[RETURN]鍵表示結束轉換工作，回至功能表畫面。

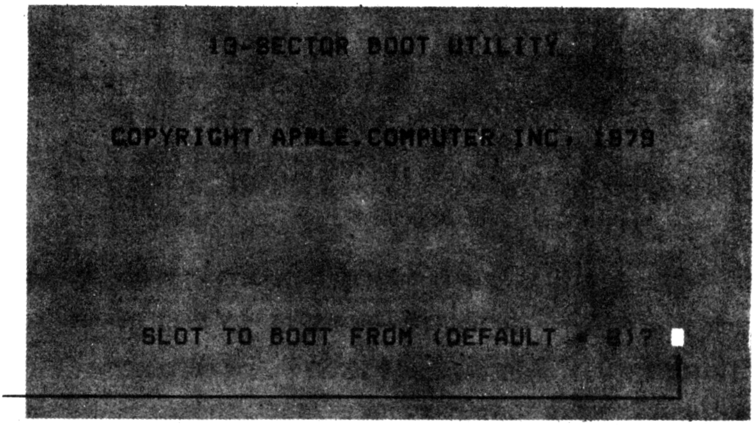
不需轉換即可執行13磁區磁碟的程式

在您的16磁區格式的電腦系統裡，您可以藉助於START13程式和 BASICS 磁碟來執行13磁區格式的磁碟內程式。START 13程式和 BASICS磁碟是專門適用於那些已被加上"拷貝保護"的13磁區磁碟。

BASICS 磁碟不再供應給DOS使用者，因爲 START13 程式比較容易使用。若您已有該磁碟，並有說明書告訴您如何"啓動BASICS 磁碟"，您現在可用 START13 程式取代之。

使用START 13程式

START13 程式可將13磁區磁碟內的資訊載入記憶體內。基本上，START13 程式是讓您的系統能啓動13磁區磁碟的特別載入程式(loader)。



圖A-3　START 13標題畫面

實　例

本例告訴您如何使用 START13 自13磁區磁碟內讀取資訊。

1. 將 **SYSTEM MASTER** 磁碟放在第一部磁碟機，並執行 **START13** 程式，打入

   [R][U][N][SPACE][S][T][A][R][T][1][3]

   並按[RETURN]鍵，EXECUTING BOOT13 訊息顯示後，即出現如圖A-3的畫面。
2. 當您看到 13-SECTOR BOOT UTILITY，並接著 SLOT TO BOOT FROM (DEFAULT=6)? 訊息時，請取出 **SYSTEM MASTER** 磁碟片並插入您的13磁區磁碟至第一部磁碟機。
3. 假若磁碟機控制界面卡已插在編號6的擴充槽，請按[RETURN]鍵。若該界面卡是插在另一擴充槽，則在

按下[RETURN]鍵之前，打入對應的擴充槽編號。
假若您的13磁區磁碟是個"自動啓動磁碟"(startup disk)，則在磁碟的啓動程式會自動立即執行之。例如，若您的磁碟含有Apple Writer 1.0，則會看到 Apple WRITE 的功能表畫面。

(註：若您用到較早期的DOS版本，您可能較習慣用BOOT 13程式。實際上，BOOT13 的功能與 START13 完全相同，START13 程式是啓動 BOOT13 的執行。)

使用 BASICS 磁碟

BASICS 磁碟同樣也可載入 13 磁區磁碟的內含資訊至記憶體內，並讓您的程式可以讀取和寫入 13 磁區格式的磁碟。當您希望用 BASICS 磁碟處理13磁區格式的檔案，您必須以BASICS磁碟來啓動系統(不可以 SYSTEM MASTER 磁碟啓動之)。

啓動 BASICS 磁碟：若您在其他早期Apple手册讀過"啓動 BASICS 磁碟"一節，您可以如下所述的步驟啓動BASICS 磁碟，或者執行 START13 程式。

在 16 磁區格式的系統上使用 13 磁區磁碟時，須將 BASICS 磁碟放在第一部磁碟機(接在擴充槽 6 號)再開啓電腦的電源。當您螢幕出現如下的訊息：

```
INSERT YOUR 13-SECTOR DISKETTE
        AND PRESS RETURN
```

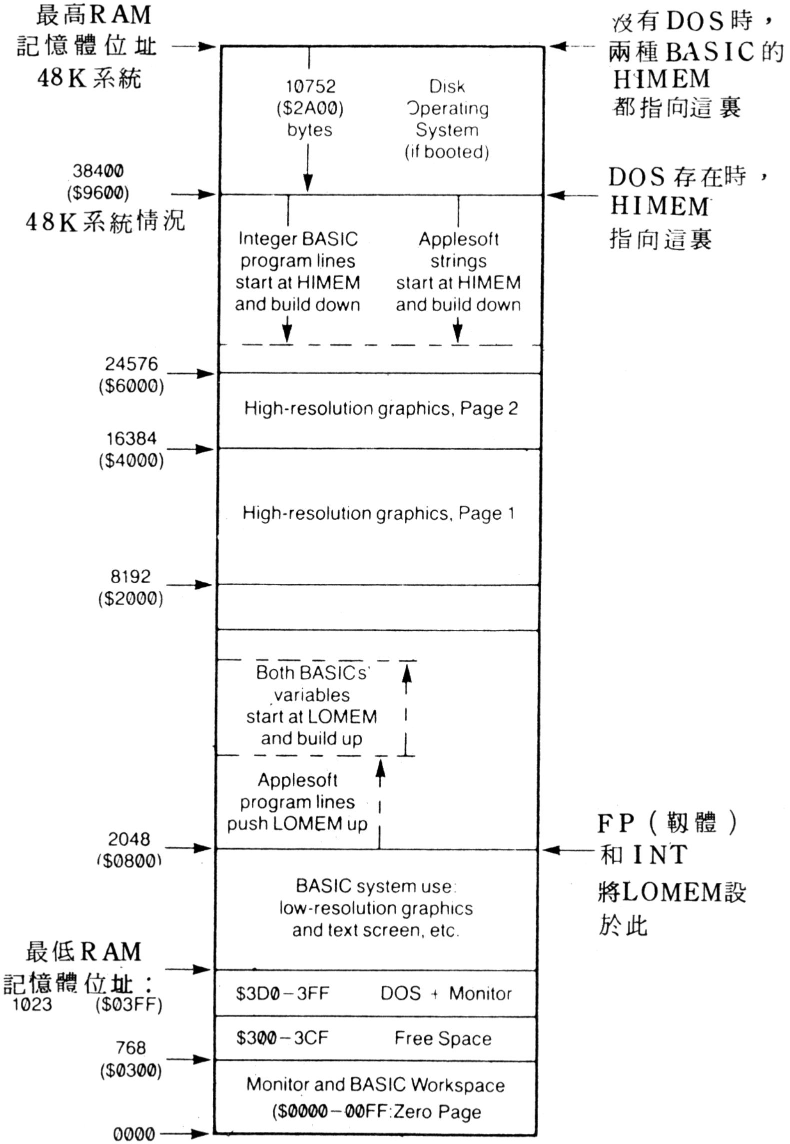
請將13磁區磁碟放入磁碟機，並按下[RETURN]鍵。

注意，BASICS 磁碟本身並未包含有DOS；假若您以BASICS 磁碟啓動了系統，嘗試使用DOS指令時，可能會在螢幕出現雜

亂的訊息。

若某一程式自動地執行，即表示您的 13 磁區磁碟是含有“轉鑰程式”（ turnkey program ）；請打入**CATALOG**查看磁碟內含有那些檔案，利用 RUN 指令執行任何您希望的程式。

圖 6 - 3 DOS 與兩種 BASIC 所使用的記憶區域



附錄B

DOS錯誤訊息

錯誤訊息（ error message ）警告您有一問題已發生，指示您下達給電腦的指令不符合 DOS 的指令規則。下列不同格式的錯誤訊息可以告訴您，是由 DOS 發出或是由 BASIC 語言發生的。

格　　式	訊息的來源
SYNTAX ERROR	DOS
?SYNTAX ERROR	Applesofe BASIC
*** SYNTAX ERR	Integer BASIC

本附錄只討論 DOS 這一類的錯誤訊息，至於 Applesoft 和 Integer BASIC 的部份，請讀者參考對應的語言使用手册。在本附錄中將詳述每一個錯誤訊息的原因，並提供修正的方法。

DOS錯誤訊息

表 B—1是 DOS 錯誤訊息的簡表，在該表之後，以“英文字母”為排列次序，分別說明錯誤的原因及解決之道。

表 B-1 DOS 錯誤訊息

訊　　息	最可能的原因
DISK FULL	檔案目錄中含有太多檔案，或磁碟上的所有磁區已用盡。
END OF DATA	讀取資料時已超出檔案結尾。
FILE LOCKED	試圖改寫、刪除或更名一已鎖住的檔案。
FILE NOT FOUND	檔名拼錯字，或指定的檔案不存在於磁碟。
FILE TYPE MISMATCH	檔案型態不符合
I/O ERROR	磁碟機的門未關好，磁碟片未啓始化，磁碟機已故障或磁碟片已磨損。
LANGUAGE NOT AVAILABLE	所要的語言不在電腦中。
NO BUFFERS AVAILABLE	開啓的文字檔個數太多。
NOT DIRECT COMMAND	必須在程式中使用該指令。
PROGRAM TOO LARGE	可用的記憶體不足。
RANGE ERROR	指令的引數值太大或太小。
SYNTAX ERROR	不正確的檔案名稱或引數值。
VOLUME MISMATCH	錯誤的卷號。
WRITE PROTECTED	沒有“可寫缺口”或被貼上了寫入保護貼紙。

DISK FULL

DISK FULL 訊息表示 DOS 要儲存資訊的磁碟時，發現沒有足夠的空間可用。在此種情況下，DOS 關閉所有檔案並保存所有可能的資訊。

當您看到 DISK FULL 訊息時，可以刪除磁碟上一、兩個檔案，然後再將記憶體中的資訊儲存於磁碟；或者更換磁碟片，再將

記憶體中的資訊存入有足夠空間的磁碟片。

假若已出現此訊息，您再嘗試儲存一檔案至磁碟時，目前磁碟目錄的某一個檔案對應的“磁區長度”會被設定成零。除了目錄的某項目會出現奇特的現象外，已存在的檔案不會受到任何損壞。

END OF DATA

END OF DATA 訊息是表示您的程式在擷取資訊當中，讀到檔案中沒有存放任何資料的區域。

任何超出“循序存取文字檔”的最後一欄或是超出“隨意存取文字檔”中任何一個資料錄的最後一欄位址，其位元組的內含值可能是0。“零”是代表 ASCII 的“null 字元”。任何指令如果要讀取 null字元，就會產生 END OF DATA 訊息。

假若在程式中不適當的使用 INPUT 或GET指述，也有可能會出現 END OF DATA 。在本訊息出現後，欲防止任何資料漏失不見，請即下達CLOSE指令。請查閱 DOS 程式設計者手册，可找到詳細的解決之道。

FILE LOCKED

FILE LOCKED 訊息是表示您正在嘗試儲存、寫入、變更、删除或更名一個已鎖住的檔案。請查驗目錄，已鎖住的檔案已標示一個星號（ * ）於檔名之旁。欲解除鎖定，使用 UNLOCK 指令。

FILE NOT FOUND

FILE NOT FOUND 表示您所指定檔名的檔案，不在目前使用的磁碟片上。下列是一些最可能的出錯原因：

- 您可能拼錯檔案名稱，輸入不正確的檔名，或者省略了隔開檔名與引數的逗號。仔細地查驗一次。看看目錄中正確的檔名。
- 檔案不在指定的磁碟內。請查驗目錄。
- 可能檔案已無意中被删除了，請查驗目錄。
- 如果每次啓動磁碟時，都會出現 FILE NOT FOUND 訊息，則您必須告訴 **DOS** 該磁碟的歡迎詞程式之檔名。假若您的磁碟片沒有任何檔案需要保留，則可以利用 **INIT** 指令啓始此磁碟。如果您已不記得磁碟中的歡迎詞程式之檔名，可以執行**MASTER** 程式加以更名。（**MASTER** 程式的用法，請參考 **DOS** 程式設計者手册。）

FILE TYPE MISMATCH

FILE TYPE MISMATCH 表示**DOS** 指令中所指定的檔案其檔案型態不適合此指令使用。

利用**CATALOG**指令檢查檔案的型態，再對照表 B—2 以確定您使用的指令及型態可互相配合。

如果您確定指令是正確的，則使用不在此磁碟的某個檔案名稱，或使用不同的磁碟片，或將既存的檔案加以更名，或删除一既存的檔案。

表 B-2　DOS 指令及適用的檔案型態

指　令	合 法 的 檔 案 型 態
LOAD, RUN, SAVE	Applesoft 或 Integer BASIC
CHAIN	Integer BASIC
OPEN, READ, WRITE, APPEND, POSITION, EXEC	文字檔（Text file）
BLOAD, BRUN, BSAVE	二進制機器碼檔案（Binary）

I/O ERROR

I/O ERROR 表示 DOS 無法由磁碟中擷取或存入資訊。在發出此訊息之前，DOS 已嘗試 96 次讀寫動作。可能的原因是：

- 磁碟機的門未關好，將它關上。
- 選定的或預設的磁碟機沒有放入磁碟片。請放入磁碟片，並關上門。
- 磁碟片可能尚未啓始化，趕快利用 INIT 指令準備一可用的磁碟。（註：請參考本書第三章的"準備一磁碟片"一節 INIT 指令。）
- 磁碟片可能沒有適當地插好，小心地從磁碟機中抽出磁碟片，再重新插入。
- VERIFY 指令已指示檔案存入磁碟有誤。
- 16 磁區版本的DOS系統意圖使用 13 磁區的磁碟，請參考附錄A。
- 磁碟機代號（Dn）引數所指定的磁碟機並不存在於系統中。目前預設的磁碟機是一個不存在的磁碟機，請在下一個

DOS 指令中指明正確的D引數。

- 擴充槽編號（Sn）引數所指定的擴充槽並無磁碟機控制界面卡。目前錯誤的槽號成為預設值，且DOS 假設這個沒有連上此槽的磁碟片仍在使用中。即使下一個DOS 指令指定正確的擴充槽，DOS 將永遠在等候這片不存在的磁碟，以便完成上個指令的動作，此情況已相當於系統"當掉"（down）了。假若記憶體內並無任何您想保存的程式，則重新啓動DOS 即可，假若要使您的程式完整地復原，則必須依循以下的步驟操作：

  1. 按 RESET 鍵。
  2. 打入 C A T A L O G SPACE S n ，其中n是正確的擴充槽編號。假若產生的結果是 SYNTAX ERROR ，表示DOS 已被切斷；須重新連接上DOS 假若出現的是 BASIC 提示符號（]或> ），請打入

     C A L L SPACE 1 0 0 2

     若出現的是監督程式的星號提示符號（ * ），請打入 3 D 0 G （請注意，它是零）
  3. 再度下達目錄指令——CATALOG, Sn ，則一切都恢復正常。

無論如何，利用VERIFY 指令確定您所使用的檔案已正確無誤存於磁碟片。如果儲存資料時發生錯誤，但資料仍在記憶中，請再嘗試儲存它們。

LANGUAGE NOT AVAILABLE

在DOS 無法找到它想使用的 Integer BASIC 或 Applesoft

BASIC 語言系統時，會發出本項訊息。FP，INT，LOAD 或 RUN 指令會自動尋找它適用的語言，可能會出現這項錯誤。

假若您下達一指令，希望由目前磁碟機上的磁碟請求使用 Applesoft 語言，而磁碟並無APPLESOFT或FPBASIC程式，即會產生此錯誤訊息。請換上存有上述程式的磁碟（例如，SYSTEM MASTER 磁碟片），然後再下指令。

當您下指令要求使用 Integer BASIC ，而 Integer BASIC 不在電腦時，也會出現此訊息。假若您的電腦有足夠的記憶體容納 Applesoft 和 Integer BASIC ，您可以由 SYSTEM MASTER 磁碟片載入 Integer BASIC 至記憶體。

當您請求使用 Applesoft 語言，DOS 先在 ROM(唯讀記憶體)或語言界面卡尋找此語言，假若沒找著，則再到 RAM （隨意存取記憶體）看看它是否存在。若仍沒找著，則 DOS 在目前預設的磁碟機的磁碟（即指定的擴充槽及磁碟機代號的磁碟）找尋 Applesoft 語言。

當您請求使用 Integer BASIC ，DOS 是在 ROM 或 RAM 尋找此語言。

假若用 SYSTEM MASTER 磁碟啓動 DOS 時，出現 LANGUAGE NOT AVAILABLE ，即表示您的電腦沒有足夠的記憶體（需64K或更多）同時容納兩種語言。

NO BUFFERS AVAILABLE

每一次開啓檔案以及每個DOS指令在執行時（PR #, IN #, MAXFILES 除外）都需要使用“檔案保留區”（ file buffer ）。NO BUFFERS AVAILABLE 即表示您或您的程式要開啓的檔案個數，或下達的DOS 指令需用的保留區已經超出記憶體中可用的保留區數量。

下達 CLOSE 指令即可釋放檔案保留區。您可以利用MAXFILES 指令來增加檔案保留區的數量。要知道如何增加或減少檔案保留區的數量，請參考 DOS 程式設計者手冊的第 5 章MAXFILES 指令。

NOT DIRECT COMMAND

NOT DIRECT COMMAND 表示您利用“立即執行”的方式下達APPEND,OPEN,POSITION,READ ，或 WRITE 指令。這些指令只能用在程式中的 PRINT 指述，請參考DOS 程式設計者手冊的 2、3、4 章說明。

當一程式已中止再嘗試重新啓動時，也可能會出現此訊息。利用下列的POKE指令可以解決這個錯誤，請打入

P O K E SPACE 5 1 , 0 : C O N T

PROGRAM TOO LARGE

PROGRAM TOO LARGE 表示DOS 要載入磁碟的檔案時，發現主記憶體沒有足夠的空間來容納此檔案的內容。

如果目前正處於“立即執行模式”，則依據您的程式型態選用 FP 或 INT 其中一個指令。當DOS 執行 FP 或 INT 時，會檢查記憶體，消除某些程式所造成的小片段，合併所有可用的空間。（註：FP 和 INT 指令是用來切換兩種不同的 BASIC 語言，請參考您的BASIC 手冊。）

執行中的程式出了問題時，也 能會出現本項訊息。可能的原因是先前的一個程式將 HIMEM: LOMEM: 的位址設定成不適

當的值，使您的程式無法載入。

您可以利用 MAXFILES 指令減少檔案保留區的數量，使您的 BASIC 程式有較多的可用記憶體。或者利用POKE 指述將正確的數值存入 HIMEM: 或 LOMEM: 的位置。要了解更多這一類程式設計的技巧，請參考 DOS 程式設計者手册。

RANGE ERROR

RANGE ERROR 表示指令中的引數值太大或太小。表B－3 列舉各引數所允許的數值範圍。

表B－3　各引數的最大值和最小值

		引數	最小值	最大值
所有檔案：	擴充槽（Slot）	S	1	7
	磁碟機（Drive）	D	1	2
	卷　號（Volume）	V	0	254
循序存取文字檔：	位元組（Byte）	B	0	32767
	相對欄（Relative field）	R	0	32767
	絕對欄（Absolute Field）	R	0	32767
隨意存取文字檔：	資料錄長度（Record Lengeh）	L	1	32767
	資料錄編號（Record Number）	R	0	32767
二進制機器碼檔案：	起始位址	A	0	65535
	位元組長度	L	1	32767
DOS指令：	PR#n	n	0	16
	IN#n	n	0	16
	MAXFILES n	n	1	16

（註：若PR#和IN# 的引數介於 8 到 16 之間，則可能產生不可預期的結果。）

使用的值如果超出表 B－3 所指定的範圍，並不一定會出現 RANGE ERROR 錯誤訊息。任何 DOS 指令若指定的引數值小於 0 或者大於 65535，可能會出現 SYNTAX ERROR 。

SYNTAX ERROR

SYNTAX ERROR 表示您或您的程式使用的指令，用錯了引數值或者不恰當的間隔字元（如逗號或空白字元）。假若指令缺了應有的引數，也會出現本訊息。檢查指令的語法格式後再試一次。

產生本錯誤的最可能原因是在輸入指令時打錯字。只要再檢查是否拼錯，即可再試一次。

VOLUME MISMATCH

VOLUME MISMATCH 表示 DOS 指令中的卷號（Vn ）與磁碟片當初啓始時所指定的卷號（即 INIT 指令）不同。利用 CATALOG指令檢查磁碟的卷號。

WRITE PROTECTED

WRITE PROTECTED 表示 DOS 無法在指定的磁碟片進行儲存、寫入或者删除資訊。“寫入保護”（Write-protected ）表示磁碟片沒有“可寫缺口”（Write-enable notch），或者該缺口已貼上了（不透光的）貼紙。下列是一些可能的原因：

- 在可寫缺口已黏上保護貼紙。只要撕去此貼紙，再下指令即可。
- 若執行拷貝程式（如 COPYA 或 COPY 程式）時出現本項

訊息，可能磁碟片沒有適當地插入磁碟機內。查驗磁碟片在磁碟機的安放位置。

- 可能磁碟片並無可寫缺口。換上另一片磁碟再儲存檔案。

停止程式的執行

有時候，您可能需要中斷執行中的程式。下列有幾種方法可以採用。遞增的數字是代表“嚴重”的程度，但同樣都能達成停止的效果。

1. 按下 ESC 鍵，它位於鍵盤的左上角。它可讓您“跳離”（escape）程式的執行狀態。
2. 按下 CONTROL-C，取消程式的執行。
3. 按下 CONTROL-C，再按 RETURN。更徹底地取消程式的執行。
4. 按 CONTROL-RESET 可停止執行中的程式，並可能回至上一個出現的功能表畫面。該程式仍存在記憶體內。
5. 若是 Apple IIe 機型，按下(蘋果鍵)再按下 CONTROL-RESET，而後反向放掉按鍵（最後是放掉(蘋果鍵)）。本法非常強制地中斷，會清除已在記憶體的任何資訊。
6. 關掉電腦電源。您已無機會再使用被您中止的程式，但這是向電腦提醒“人才是電腦的老闆！”最好的方式。注意，它徹底清除任何存在記憶體的程式或資訊。

注意，在本書中兩個字鍵之間的“橫劃”（hyphen　是表示您必須“同時按住”兩個字鍵。例如 CONTROL-RESET 即是同時按下 CONTROL 和 RESET 鍵。不過，實際上，

您可以先按下 CONTROL ，暫時不要放掉它（即按住它），再按下 RESET ，即相當於"同時按住"的效果。

附錄C

DOS磁碟內的各種程式

本附錄列舉在SYSTEM MASTER 和SAMPLE PROGRAMS 磁碟內各種程式，並簡介各程式的作用。

SYSTEM MASTER磁碟內的程式

本節簡略地說明 SYSTEM MASTER 磁碟內各程式的作用。假若某程式在本手册曾使用過，則在內文中加註“加號”（＋）。要查尋某一程式曾在何處討論過，請利用“目錄”找尋頁數。假若某程式並未有＋號標誌，表示它是包含在“ DOS 程式設計者手册”裡。

HELLO	— Applesoft 型態的歡迎詞程式，祇要在 DOS啓動後，而Applesoft BASIC 語言已可用之情況下，即自動執行本程式。(+)
APPLESOFT	—Integer BASIC 型態的歡迎詞程式，祇要在 DOS啓動後，而Applesoft BASIC 語言已知無法使用之情況下，即自動執行本程式。

BOOT13	一機器碼程式，用在載入"13 磁區" 舊型式的磁碟，該磁碟是利用較早期的DOS程式(DOS 3.3)產生的。
CHAIN	一機器碼程式，它可以載入並執行第二個程式，但不破壞(清除)已存於記憶體內，第一個程式的變數及陣列的內含值。兩個要聯接(chained)的程式其型態必須是Applesoft BASIC。
CONVERT13	一Applesoft型態之程式，它執行MUFFIN程式，將一13磁區的磁碟轉換成16磁區格式的磁碟。(+)
COPY	在使用Integer BASIC 語言情況下，利用本程式來複製整片磁碟。(+)
COPY.OBJ0	一機器語言的常式(routine)，被COPY或COPYA程式所叫用。
COPYA	在使用Applesoft BASIC語言情況下，利用本程式來複製整片磁碟。(+)
FID	一機器碼程式，合聚DOS提供的幾種指令功能的程式。
FILEM	一機器碼程式，它可啓動FID的執行，您可以一次同時操作數個檔案。(+)
FPBASIC	存於磁碟片的Applesoft BASIC 語言程式，其型態是機器碼。(+)
INTBASIC	存於磁碟片的Integer BASIC 語言程式，其型態是機器碼。
LOADER.OBJ0	存在SYSTEM MASTER 磁碟上，經由HELLO載入記憶體的機器語言程式。LOADER. OBJ0 的作用是將另一套BASIC

	語言載入語言界面卡或者記憶體內。假若已存在著第二套語言系統，則不發生效用。
	LOADER. OBJ0 同時將存於ROM的監督程式，轉存至語言界面卡或記憶體的 $F800 位址上。
MASTER	一個啓動MASTER CREATE程式（其型態是機器碼）的Applesoft BASIC 程式，將一已啓始化的磁碟片轉變成"主"磁碟（master disk）。
MASTER CREATE	一機器碼程式，可產生一主磁碟，亦即是內含有可將 DOS 本身程式儘量放在記憶體中較高位置的 DOS 系統磁碟。一主磁碟可適用於任何記憶體容量的 Apple II 電腦。
MUFFIN	可轉換"13磁區"磁碟至"16磁區"格式的磁碟之機器碼程式。
RENUMBER	一Applesoft 程式，可將一BASIC 程式或合併的兩個程式其列號（line number）重新加以編號。
SLOT#	一Applesoft程式，可回報目前的擴充槽及磁碟機預設值。
START13	一Applesoft 程式，可啓動BOOT13 程式之程式，其目的是啓動先前版本DOS產生的13磁區磁碟。

SAMPLE PROGRAMS磁碟內的程式

本節簡略地說明SAMPLE PROGRAMS 磁碟內各程式的作用

。在本書中曾使用它們來展示 DOS 指令的用法，若曾在本書討論過，則下文中加註“加號”（＋），至於詳細的頁數，請查尋目錄表。假若並未加註＋號標誌，表示它是包含在“DOS 程式設計者手册”裡。

HELLO	Applesoft 型態的歡迎詞程式（greeting program）。（＋）
ADDRESS	用 Applesoft BASIC 撰寫的示範用程式，用以闡示“隨意存取檔案”的讀取和寫入動作。
ANIMALS	用 Integer BASIC 撰寫的遊戲程式，可用來建立動物名稱的資料檔案。
APPLE PROMS	RANDOM程式須用到的資料檔案，它含有有關於物料庫存的資料錄。
APPLEVISION	用 Integer BASIC 撰寫的示範用程式。（＋）
BLACK BOOK	儲存ADDRESS程式中須用到的資料錄之資料檔案。
BRICK OUT	以 Applesoft BASIC 撰寫的遊戲程式。請利用鍵盤的方向鍵操作本遊戲，或搖桿來控制方向。（＋）
COLOR TEST	以 Applesoft BASIC 撰寫的示範用程式，可藉它調整彩色電視的螢幕特性。（＋）
DELETE.ME.1	Applesoft 示範用程式，讓您練習刪除一個檔案。（＋）
DELETE.ME.2	Applesoft示範用程式，讓您練習刪除一個檔案。（＋）
DELETE.ME.3	Applesoft示範用程式，讓您練習刪除一個

	檔案。（＋）
EXEC DEMO	Applesoft示範用程式，告訴您一個EXEC程式檔案是如何產生的，及如何執行它。
FPBASIC	存於磁碟的Applesoft BASIC 語言程式，機器碼檔案。（譯註：FPBASIC即是習稱的浮點式BASIC Interpreter）
GET TEXT	Applesoft示範用程式，可讀取一文字檔的內容。
HELLO	以Applesoft撰寫的歡迎詞程式。
INTBASIC	存於磁碟的Integer BASIC語言程式，機器碼檔案。
LOADER.OBJ0	在SYSTEM MASTER 磁碟上，經由HELLO 載入記憶體的機器語言程式。LOADER.OBJ0 的作用是將另一套BASIC語言載入語言界面卡或者記憶體內。假若已存在著第二套語言系統，則不發生效用。LOADER.OBJ0 同時將存於ROM的監督程式，轉存至語言界面卡或記憶體的$F800 位址起的區域。
LOCK.ME.1	Applesoft示範用程式，讓您練習鎖定一尚未鎖住的檔案。（＋）
LOCKED.UP.1	Applesoft示範用程式，讓您練習解除一已鎖定檔案的鎖住保護。（＋）
LOCKED.UP.2	Applesoft示範用程式，讓您練習解除一已鎖定檔案的鎖住保護。（＋）
MAKE TEXT	Applesoft示範用程式，闡示一循序文字檔的建立過程。
ONERR DEMO	展示“錯誤復原”（error recovery）技

	巧的樣本程式。它檢查某一檔案是否已設定鎖住保護，若是的話，則解除其鎖定狀態。
POKER	Applesoft示範用程式，闡示機器語言程式轉換成BASIC程式的處理過程。
PHONE LIST	Applesoft示範用程式。（＋）
RANDOM	Applesoft示範用程式，闡示隨意處理文字檔的讀取和寫入動作。它利用APPLE PROMS記錄使用到的資料。
VERIFY.ME	Applesoft示範用程式，讓您練習查證一檔案是否已正確地存入磁碟內。（＋）

附錄D

DOS操作概念及指令摘要

本附錄只針對“DOS使用者手册”所涵蓋的操作概念、指令及程式作個精簡的摘要整理。若希望更進一步的說明，請參考“DOS程式設計者手册”。

DOS操作概念

DOS可在Apple II 電腦的所有機種上運轉，包含Apple II，Apple II Plus以及Apple IIe 。若有特別適用於某種機型的資訊流程和操作程序，將會特別於適當章節中明示。

啓　動

“啓動”是將您的電腦扭開電源開關的操作過程（或者模擬的相當過程），以便讓DOS 載入記憶體內（請參考第1章）。

1. 將DOS磁碟片插入第一部磁碟機（Drive 1）。
2. 開啓顯示裝置的開關，如電視機或視訊監視器（video monitor，或稱顯示器）。
3. 找到電腦的電源開關，並打開它。

對於 Apple IIe 機型而言，您可以在電源已通之下模擬啓動的過程。先按住 [] ，再按下 [CONTROL]-[RESET] 。放掉 [RESET] ，[CONTROL] 鍵，最後鬆開 [] 。

重新啓動 DOS

“重新啓動”代表的意義是在電源已開啓情況下，重新“連接”DOS 的操作過程。某些重新啓動的方法並不會影響到已存於主記憶體內的資訊，有些則會破壞其內的資訊。下列方法其可能的毀損程度是依編號逐漸昇高的：

1. 按下 [CONTROL]-[RESET] 。本法並不影響到主記憶體的內容。假若您的系統已“當掉”，請首先用本法試試看。
2. 打入 [C][A][L][L][SPACE][1][0][0][2]。它是 BASIC 指令，可以重新連接 DOS 但不影響到主記憶體的內容。
3. 打入 [P][R][SHIFT]-[#][6] 自第一部磁碟機（擴充槽 6）重新啓動 DOS ，適用於 Apple II 的所有機種。若是 Apple IIe 尙可以按下 []-[CONTROL]-[RESET] 。啓動 DOS 。兩種方式均會破壞已存於記憶體的任何資訊。本法是用來測試一新系統磁碟是否“可啓動”的最好方法。
4. 若螢幕上出現的是 *（星號），表示在監督程式之控制下，打入下列任何一種：

[6][CONTROL]-[P]

[6][CONTROL]-[K]

[3][D][0][G]（注意，內含一個數字零，不是英文字母 O ）

[C][6][0][0][G]（注意，內含二個數字零）

如果您的磁碟機控制界面卡是插在擴充槽 6 以外的其他擴充槽，請以對應該擴充槽的編號代入上列的 6。

磁碟容量

DOS 使用者最多可使用到 496 個磁區（ sector ），每一磁區相當於儲存 256位元組（ byte ）的資訊。

最小的檔案長度，如一空的文字檔（ empty text file ）是 1 個磁區（內存有該檔的磁軌／磁區的記錄項），如一空的Applesoft BASIC 檔案，Integer BASIC 以及機器碼檔案（ binary file 則是 2 個磁區（其中一個內存磁軌／磁區的記錄項，另一個則記錄程式長度，是程式所在的第一個磁區）。

記號說明

在本書中指令語法（ syntax ）中出現的符號（代號）與其意義列舉如下：

UPPERCASE	大寫字體，表示是 DOS 指令的眞實名稱，或其他您須依字面眞實打入的名稱。
lowercase	小寫字體，表示由您打入的任何適當名稱，例如一程式的名稱。
fn	表示由您打入的檔案名稱。
[]	括弧內項目代表它是指令中可有可無的選擇項（ option ）引數。您可依據情況，在指令中選用或略去本項，不管如何，請不要打入括弧本身。
	代表您希望的數值。

語 法

“檔案名稱”通常是一指令名稱之後的第一個引數（ argument ），其餘的引數其排列次序之前後關係無關於指令的效用，各引數之間須以逗號分隔之，例如在檔案名稱之後加上逗號以分離其他後隨的引數（參考第 2 章）。如下例，

```
INIT fn [,Vn] [,Sn] [,Dn]
```

指令語法中各項的說明如下所示：

- 指令名稱 INIT 是以大寫字體表示之，請完全忠實地一字不漏的鍵入該名稱。
- fn 項引數代表“ filename ”（檔案名稱），請在下達本指令時，務必加入您希望的檔案名稱。
- Vn 項引數是可有可無的選擇項，它代表“ volume number ”（卷號）。當您選用本項時，請打入V（因它是大寫字體），而後再打入您希望的數目。
- Sn 項引數是可有可無的選擇項，它代表“ slot number ”（擴充槽編號）。當您選用本項時，請打入S（因它是大寫字體），而後再打入您希望的數目。
- Dn 項引數是可有可無的選擇項，它代表“ drive number ”（磁碟機代號）。當您選用本項時，請打入D（因爲它是大寫字體），而後再打入您希望的數目。

引 數

下列是在 DOS 指令中出現的各引數之意義簡述。詳細內容

請參考第2章。

- fn代表" filename "。
  一檔案之名稱可以由1個到 30 個字元構成，但必須以英文字母開頭。除了逗號之外的任何可印出（顯示）的文字均可以是檔名的一部份。
- n代表" number "
  您可以用一整數或十六進制數目代入此記號。
- D代表" drive "
  本項引數必須後接一個數目，1或是2，它代表您希望使用的磁碟機代號。此代號原先是預設爲1，而後是設定爲最近使用過的指令中D項引數的數目。
- S代表" slot "
  本項引數必須後接一個數目，1到7之間的數字，它代表您希望使用的擴充槽編號。此編號原先是在啓動DOS 時該擴充槽的編號，而後是設定爲最近使用過的指令中S項引數的數目。
- V代表" volume "
  本項引數必須後接一個數目，0到254 之間的任何一數。它代表您希望的磁碟片卷號，它原先是啓動DOS 的磁碟片所屬的卷號，而後是設定爲最近使用的V項引數數值或者讀取磁碟的卷號。當在使用 INIT 指令時略去本項，則系統假設該磁碟是卷號 254。本項引數很少選用它。

指令摘要

本節僅列舉曾出現於本書的DOS 指令，至於"所有"的指令

解說，請見“ DOS 程式設計者手册 ”。下列各指令是以英文字母排列的，同時在說明中加註解說該指令的章節編號。

CATALOG

語法：CATALOG[,Sn][,Dn]
例　：CATALOG

本指令可以顯示所指定的磁碟片之內容的一覽表（目錄），它顯示卷號及磁碟片各檔案之檔名、檔案型態、佔用的磁碟空間大小（其單位是“磁區”，sector ）並標明各檔是否已鎖定；假若有一星號（ * ）在旁，即表示該檔已被鎖定（ lock ）。

以下是各種檔案型態的說明：

A　Applesoft BASIC 程式檔案，以 SAVE 指令產生的。
B　二進碼（ binary ）記憶體映像（ memory-image ）檔。以 BSAVE 指令產生的。（譯註：本書譯名取“機器碼檔案”）。
I　Integer BASIC 程式檔案，以 SAVE 指令產生的。
R　可重定位（ Relocatable ）的組合語言程式檔（請參考“6502 組合語言／DOS工具集”（ 6502 Assembler / DOS Tool Kit ）。
S　保留，為以後之用。
T　文字檔（ text ），以 OPEN 指令配合WRITE指令產生的。

若某檔案的長度超過 255 磁區，則 CATALOG 顯示的目錄在檔案長度部份是超過 255 後自零起算的部份。它可能會給予您錯誤的檔案長度印象，因而錯估了剩餘的磁碟空間。

詳見第3章。

DELETE

語法：DELETE　fn[,Sn][,Dn][,Vn]
例　：DELETE　TEST

本指令用來刪除磁碟上的某一檔案。該檔案必須是尚未鎖定。當指名的檔案並未存在於磁碟上時，DOS 顯示錯誤訊息 FILE NOT FOUND 。

詳見第4章。

IN#

語法：IN#
例　：IN#6

本指令自指定的擴充槽上所連接的裝置上讀取資訊（通常是從鍵盤上讀入資訊，本指令用來改變資訊讀入的路徑。）。#字元是指令名稱的一部份，請勿遺漏它。

利用IN #0可將DOS 讀取資訊的流程回復至鍵盤讀入的狀況。

假若在您指定的擴充槽上並無控制界面卡，則DOS 似乎已無效用，您只好重新啓動電腦（請參考本附錄的第一部份）。

詳見第5章。

INIT

語法：INIT fn[,Sn][,Dn][,Vn]

例　：INIT HELLO,V18

本指令用來啓始化（initialize）一磁碟片，將磁碟片規劃成數十個磁軌和磁區以準備接受資訊，同時並將 DOS 系統程式及一歡迎詞程式（您指名的 fn 檔案）存入該磁碟內。

詳見第 3 章

警告： INIT 指令將會完全清除已存於磁碟的任何東西，請在使用本指令之前，確定該磁碟的內含資訊已不再有利用價值。

LOAD

語法：LOAD fn[,Sn][,Dn][,Vn]
例　：LOAD DOW JONES,S6,D1

本指令自磁碟片拷貝出您指定的檔案（其型態是 A 或 I）並存入主記憶體內。一旦該程式已存於記憶體中，您就可以執行它，修改它或者再儲存於磁碟。

在載入新程式至記憶體內，DOS 先關閉所有已開啓的檔案，清除既存於記憶體的內容，並切換至您指名的檔案所屬的檔案型態至 BASIC 語言。

假若檔案型態是 A，而 Applesoft BASIC 並不是目前的 BASIC 語言系統（既不存在於記憶體也不在韌體內），則 DOS 自動在磁碟上找尋 Applesoft BASIC 語言程式，如果磁碟上並無該語言程式，或者您的電腦無足夠位置容納兩套 BASIC 語言，DOS 顯示 LANGUAGE NOT AVAILABLE 錯誤訊息。

若檔案型態是 I，DOS 先在 RAM（語言界面卡的 RAM 部份）尋找 Integer BASIC 語言程式，若尋不到，再嘗試在 ROM

上尋找。一旦 DOS 無法尋著或載入您希望的語言系統，則顯示 LANGUAGE NOT AVAILABLE 訊息。

詳見第 5 章

LOCK

語法：LOCK　fn[,Sn][,Dn][,Vn]
例　：LOCK LOVE LETTERS D2

本指令可以預防一檔案在無意中被修改或刪除、更名。在目錄中指名的檔案其旁會加註星號（ * ）表示它已鎖住。除非您解除該檔案的鎖定狀態，您無法將該檔案內容加以更改，或者刪除它、更名它。

詳見第 4 章。

PR#

語法：PR#n
例　：PR#6

本指令將原本送至顯示裝置的資訊，改道傳送至您指定的擴充槽（編號是 n ）上裝置，它可能是印表機。#字元是指令名稱的一部份，請勿遺漏它。

利用 PR #0 可回復資訊送至顯示螢幕的狀態。假若您的 Apple II 電腦是 Apple IIe 機種附加 80 格文字顯示界面卡，請使用PR#3 指令回復資訊傳送至螢幕的正常狀態。

假若在您指定的擴充槽上並無控制界面卡，則 DOS 似乎已無效用，您只好重新啓動電腦（請參考本書附錄的第一部份）。

詳見第 5 章。

RENAME

語法：RENAME fn1,fn2[,Sn][,Dn][,Vn]
例 ：RENAME CURRENT, NEW, S5, D1

本指令可將原名為 fn1 的檔案更名為新檔案 fn2 。檔案的內容不受任何影響。RENAME 指令並不查驗磁碟上是否已有名稱為 fn2 的檔案。

詳見第 4 章。

RUN

語法：RUN fn[,Sn][,Dn][,Vn]
例 ：RUN ANNUITY,D2

本指令可自磁碟片拷貝出一 BASIC 程式 fn（其型態為A 或 I）存入記憶體內，並立即執行它。

詳見第 5 章。

SAVE

語法：SAVE fn[,Sn][,Dn][,Vn]
例 ：SAVE COLOR DEMOS,D2

本指令將目前存於記憶體內的 BASIC 程式以 fn 名稱儲存至磁碟內。若它是 Applesoft BASIC 程式，以A型態儲存之；若

它是 Integer **BASIC** 程式，則以 I 型態儲存之。假若磁碟已含有一相同檔名及型態的檔案，則 **DOS** 會將目前已存於記憶體的程式覆蓋寫入於既存檔案的部份。假若磁碟是含有相同檔名但不相同型態的檔案，則顯示 FILE TYPE MISMATCH 錯誤訊息。

詳見第 5 章。

UNLOCK

語法：UNLOCK fn[,Sn][,Dn][,Vn]
例　：UNLOCK RECIPES,D2

本指令用來解除一檔案的鎖定保護狀態，讓該檔案可以被更名、刪除或者變動其內容。

詳見第 4 章。

VERIFY

語法：VERIFY fn[,Sn][,Dn][,Vn]
例　：VERIFY SAM

本指令用來查證一檔案是否已正確地寫入磁碟片內。其測試方法是看看 **DOS** 能否自磁碟內讀取指名的檔案內容，順利的存入電腦的記憶體內。假若 **DOS** 能讀取檔案的內容，**DOS** 並不會顯示“已確認無誤”的訊息，但假若無法讀取該檔案，則顯示 I/O ERROR 訊息。您可以查證任何型態的檔案。

詳見第 4 章。

第二部分

程式設計手冊

目　　錄

譯序 …… IX

第 1 章　簡　介 …… 3

1.1　DOS 操作環境 …… 3

1.1.1　DOS 的磁碟 …… 4

1.1.2　使用 DOS 的要求 …… 4

1.1.3　使用說明 …… 5

1.2　如何敍述 DOS 指令 …… 5

1.2.1　表示法 …… 5

1.2.2　語　法 …… 6

1.3　使用 DOS 指令 …… 9

1.3.1　CATALOG 指令 …… 9

1.3.2　INIT 指令 …… 10

1.3.3　RENAME指令 …… 11

1.3.4　LOCK指令 …… 11

1.3.5　UNLOCK 指令 …… 12

1.3.6　DELETE 指令 …… 13

1.3.7　VERIFY 指令 …… 13

1.3.8　RUN 指令 …… 15

1.3.9　LOAD指令 …… 16

1.3.10　SAVE指令 …… 17

第 2 章　DOS的程式工具 …… 21

2.1 使用 DOS 指令的方法 …… 21
2.1.1 立即執行模式 …… 22
2.1.2 延遲執行模式 …… 22
2.1.3 指令檔 …… 22
2.2 BASIC 程式檔 …… 23
2.2.1 利用 FP 和 INT 指令切換語言 …… 25
2.2.2 由監督程式回到 BASIC …… 27
2.3 程式中使用 DOS …… 28
2.3.1 範 例 …… 29
2.3.2 注意事項 …… 30
2.4 利用MON和NOMON偵錯 …… 32
2.4.1 MON 指令 …… 33
2.4.2 NOMON指令 …… 33
2.5 利用 Applesoft 來攔截錯誤 …… 34
2.5.1 範 例 …… 35
2.5.2 找出檔名中的隱藏字元 …… 37
2.6 在程式中利用其他週邊設備 …… 38
2.6.1 用PR# 和 IN# 啓動程式 …… 40
2.6.2 PR# 指令 …… 40
2.6.3 IN# 指令 …… 41
2.7 RENUMBER程式 …… 42
2.7.1 RENUMBER指令 …… 43
2.7.2 RENUMBER的錯誤訊息 …… 48
2.8 摘 要 …… 50
2.8.1 指 令 …… 50
2.8.2 程 式 …… 51

第 3 章 循序存取文字檔的使用 …… 55

3.1 比較循序與隨意存取文字檔 …… 55
3.1.1 選擇循序或隨意存取文字檔 …… 57
3.1.2 目前位址指標 …… 58
3.2 循序存取文字檔 …… 59
3.2.1 資料欄 …… 59
3.2.2 輸入和讀取文字 …… 63
3.2.3 程式：輸入和讀取文字 …… 72
3.3 循序存取檔的指令 …… 77
3.3.1 OPEN指令——延遲執行模式 …… 77
3.3.2 CLOSE 指令——立即或延遲執行模式 …… 79
3.3.3 WRITE 指令——延遲執行模式 …… 80
3.3.4 READ指令——延遲執行模式 …… 81
3.3.5 POSITION 指令——延遲執行模式 …… 82
3.3.6 APPEND 指令——延遲執行模式 …… 83
3.4 指令摘要 …… 84

第4章 隨意存取文字檔的使用 …… 89

4.1 隨意存取文字檔 …… 89
4.1.1 資料錄長度 …… 90
4.1.2 寫入資料錄 …… 91
4.1.3 讀取資料錄 …… 92
4.2 示範：RANDOM程式 …… 92
4.3 隨意存取程式示例 …… 94
4.3.1 控制程式 …… 94
4.3.2 儲存資料錄 …… 95
4.3.3 寫入資料錄 …… 96
4.3.4 讀取資料錄 …… 98
4.4 隨意存取文字的指令 …… 100

4.4.1 OPEN 指令——延遲執行模式…………………… 100
4.4.2 CLOSE 指令——立即或延遲執行模式……… 102
4.4.3 WRITE 指令——延遲執行模式……………… 102
4.4.4 READ指令——延遲執行模式…………………… 104
4.5 指令摘要……………………………………………… 105

第五章 高級程式技巧 …………………………………… 109

5.1 指令檔之使用………………………………………… 109
5.1.1 EXEC示範…………………………………………… 110
5.1.2 建立指令檔………………………………………… 112
5.1.3 特殊用途的指令檔………………………………… 115
5.2 EXEC指令……………………………………………… 118
5.3 MAXFILES 指令……………………………………… 120
5.3.1 保留區的資訊……………………………………… 120
5.3.2 範 例……………………………………………… 122
5.4 CHAIN ………………………………………………… 122
5.4.1 CHAIN 指令——Integer BASIC ………… 122
5.4.2 CHAIN 程式——Applesoft ……………… 125
5.5 MASTER 程式………………………………………… 126
5.5.1 範 例……………………………………………… 127
5.6 製作轉鑰式磁碟……………………………………… 129
5.6.1 建立轉鑰式磁碟…………………………………… 129
5.6.2 建立主轉鑰式磁碟………………………………… 130
5.6.3 萬用磁碟…………………………………………… 130
5.7 摘 要………………………………………………… 134
5.7.1 指 令……………………………………………… 134
5.7.2 程 式……………………………………………… 135

第6章 有效使用記憶體 …… 139

6.1 二進制機器碼檔案 …… 139
6.1.1 二進制機器碼位址 …… 140
6.1.2 An與Ln：記憶位址參數 …… 141
6.2 二進制機器碼指令 …… 141
6.2.1 BRUN指令 …… 143
6.2.2 BLOAD指令 …… 144
6.2.3 BASAVE指令 …… 145
6.3 DOS 與監督程式 …… 147
6.3.1 監督程式的輸入與輸出暫存器 …… 148
6.3.2 DOS 的輸入與輸出暫存器 …… 150
6.4 記憶體的使用與進入點 …… 152
6.4.1 記憶體中的PR#和IN# …… 152
6.4.2 記憶體配置圖 …… 154
6.4.3 DOS 的進入點 …… 158
6.5 二進制機器碼檔案指令摘要 …… 161

附錄A 處理13磁區格式的磁碟 …… 167

A.1 轉換 13磁區的磁碟：CONVERT13程式 …… 168
A.1.1 實 例 …… 168
A.1.2 通配字元 …… 171
A.2 使用未轉換的 13 磁區磁碟 …… 172
A.2.1 使用START13 程式 …… 172
A.2.2 使用BASICS磁碟 …… 173

附錄B 儲存資訊的內部處理過程 …… 175

B.1 磁軌與磁區 …… 175

B.2 檔案磁區的內容 …… 177
B.2.1 各類檔案型態所用的檔案磁區格式 …… 177
B.2.2 磁軌/磁區表(track/sector list) …… 179
B.2.3 磁碟檔案名錄(disk directory) …… 180
B.2.4 卷目次表(Volume Content of Table) …… 184
B.2.5 磁軌的位元配置表(track bit map) …… 186
B.3 磁軌與磁區的分配 …… 188
B.4 為磁軌與磁區定址:RWTS副程式 …… 189
B.4.1 範 例 …… 190
B.4.2 格式:輸出入區段與設備特性表 …… 192
B.5 第3分頁的 DOS 向量 …… 194
B.6 第0分頁的使用情形 …… 196

附錄 C DOS錯誤訊息 …… 199

C.1 DOS 錯誤訊息 …… 200
C.2 修正錯誤 …… 201

附錄 D 程式 …… 213

D.1 SYSTEM MASTER 磁碟內的各種程式 …… 213
D.2 SAMPLE PROGRAMS 磁碟內的各種程式 …… 215

附錄 E DOS操作概念及指令摘要 …… 219

E.1 操作概念 …… 219
E.1.1 冷啟動 …… 219
E.1.2 溫啟動(所有Apple II 機種) …… 220
E.1.3 啟始化一磁碟片 …… 220
E.1.4 產生一主磁碟 …… 221
E.1.5 解譯 DOS 指令 …… 222

E.1.6 磁碟容量 …… 222
E.1.7 檔案型態 …… 223
E·2 指令記號之說明 …… 223
E·2.1 記號意義 …… 223
E.2.2 語法 …… 223
E·2.3 引數 …… 224
E·3 指令摘要 …… 227
E·3.1 存取磁碟內容的非程式化指令 …… 228
E.3.2 控制程式環境的指令 …… 232
E.3.3 循序存取文字檔的程式化指令 …… 235
E.3.4 隨意存取文字檔的程式化指令 …… 238
E.3.5 二進制機器碼檔案的指令 …… 241
附錄 名詞解釋 …… 243
附錄 快速提示表 …… 261

譯序

本書共分兩大部份（DOS User's Manual 和 DOS Programmer's Manual）是 Apple 電腦公司新出版的實用級參考手册，將先前的 DOS Manual（1980 年版）重新編輯改寫，以配合不同層次的使用者。本書的“DOS使用者手册”部份提供了最基本的DOS指令使用法及拷貝檔案及磁碟的操作過程，最重要的是按著章節的進展，提出電腦應用中最常用的術語，並詳加解說各術語的意義讓讀者深入了解各種 DOS 的實質作用，眞正地讓初入 Apple 大門的使用者在親切的學習中獲得了基本而必要的資訊常識。第二部份“DOS程式設計者手册”，則是全部 DOS 指令的詳加解說，以及隨Apple 主機附送的 SYSTEM MASTER 磁碟內各種程式的用法；在此書中您可以了解如何於 BASIC 程式中引用 DOS 指令，去處理順序存取和隨意存取文字檔的應用，讓您能發揮 Apple 的效益。編者以爲此新版書籍編排已大幅改進，題材重新調整，條理分明，必能有益於希望善用 Apple 電腦及 Applesoft 語言的人。

固然在台灣的蘋果熱潮已漸漸過去，然而目前的 16 位元個人電腦價錢無法普遍化尚未形成氣候，假若您擁有一部Apple 電腦，雖然它可能是“價廉”，但它確實是“多功能”的個人電腦，它的教育性功能不容置疑，其可用性也因衆多套裝軟體成功的推廣而證實，爲何我們不再重視它，重加評估它的價值呢？希望本書能讓讀者再深入瞭解 DOS 的種種，也可作爲迎接未來的新生代電腦的準備。

編者

1984,3,25

1 簡　介

1.1　DOS 操作環境　3
1.1.1　DOS 的磁碟　4
1.1.2　使用DOS 的要求　4
1.1.3　使用說明　5
1.2　如何敘述DOS 指令　5
1.2.1　表示法　5
1.2.2　語法　6
1.2.2.1　預設值　6
1.2.2.2　數目　7
1.2.2.3　槽號選擇項　7
1.2.2.4　磁碟機代號選擇項　8
1.2.2.5　卷號選擇項　8
1.2.2.6　16進位數字　8
1.3　使用DOS 指令　9
1.3.1　CATALOG 指令　9
1.3.2　INIT 指令　10
1.3.3　RENAME 指令　11
1.3.4　LOCK 指令　11
1.3.5　UNLOCK 指令　12
1.3.6　DELETE 指令　13
1.3.7　VERIFY 指令　13
1.3.8　RUN 指令　15
1.3.9　LOAD 指令　16
1.3.10　SAVE 指令　17

第1章
簡　介

1.1 DOS操作環境

本書（DOS Programmer's Manual）描述如何使用DOS指令，將程式和其他資訊由磁碟中讀出或寫入。DOS是Apple II、Apple II Plus、Apple IIe電腦的磁碟作業系統（disk operating system）。

只要將 DOS 程式引入電腦，則等於在原有的計算功能中，再加上 DOS 的指令與功能。

以下是如何善用 DOS 手册的幾點建議：

- 如果您才剛開始使用Apple II電腦，您必須再瞭解其他相關的介紹性資料。
- 在開始使用磁碟之前，最好能瞭解相關的程式規劃原理，請參考Applesoft Tutorial 和Applesoft BASIC Programmer's Reference Manual 。
- 依您對電腦和程式技巧的瞭解程度，在開始寫程式使用磁碟之前，您可能需要閱讀 DOS 使用者手册（DOS User's Manual）。DOS使用者手册對本書中複雜的內容

有簡要的介紹。

1.1.1 DOS的磁碟

標註 DOS 3.3 SYSTEM MASTER 的磁碟片即含有DOS程式本身以及其他有關磁碟使用的程式。不少這類的程式在本書中皆有介紹。

標有DOS 3.3 SAMPLE PROGRAMS的磁碟片存有示範程式，當您學習 DOS 的觀念時，可加以執行，其中也包括寫程式使用磁碟的BASIC 範例。本書將一一說明這些示範程式和其他的例子。

1.1.2 使用DOS的要求

DOS 操作系統可在任何具有16K （16 千個位元組，16 Kilobytes ）**隨意存取記憶體**（RAM, Random-Access Memory）。DOS 本身佔用大約10.5K 的記憶體。您的電腦至少要擁有32K 的記憶體，才容易執行您的程式。如果您打算同時使用兩種 BASIC 程式語言——Applesoft BASIC 和 Integer BASIC ，則您必須擁有64K 的記憶體。

您的Apple II電腦必須至少有一部磁碟機連上主機，裝設磁碟的方法請參考您的磁碟機附帶的手册。

DOS 3.3 將磁碟的空間劃分成 16 個磁區（sectors）。

註: 本章摘要介紹DOS User's Manual的資料。

註: 附錄A 說明如何利用 16 磁區（16-sector）DOS 系統來使用 13 磁區（13-sector） 的磁碟。

1.1.3 使用說明

Apple II這個名稱在本書中代表 Apple II電腦的所有機型：標準的Apple II、Apple II Plus和Apple IIe。如果某些資料僅適用特定機型，則會特別聲明。

重要名詞第一次使用時，以粗體字出現，並加以解釋，這些名詞列於書後“名詞解釋”附錄裏。

DOS 手册利用兩種醒目的段落，來提示重要資訊：**順便一提**：此方塊提供更快或更好的方式處理某些事，通常以「註」、「順便一提」等諸如此類的標題開頭。

警告：警告方塊指出危及檔案、磁碟或系統的動作，方塊內的資訊說明危險之所在，並提供規避的方法。

如果您已讀過 DOS User's Manual ，您可以跳過本章其他部份，此部份只是摘要。

1.2 如何敘述 DOS指令

DOS 指令各部份的語法（syntax）（順序和格式）都以簡要的方式來表達，如下節所示。

1.2.1 表示法

以下的表示法用來描述指令的語法：

UPPERCASE	用來表示某些事物的眞正名稱，必須直接依照指示的字元一一輸入，例如各個 DOS 指令名稱均以此種方法表示之。
lowercase	表示由您提供的事項，如程式名稱。

fn　指示您應提供的檔案名稱，最多含有 30 個字元，第一個字元必須是字母；其他的可以是任何字元，包括空白字元（space），但逗號（comma ,）除外。

[ ]　方括號所括的是指令中的選擇性引數（optional arguement）。如果您選擇此項引數時，千萬不要打入方括號，此記號只是用來表示這是選擇項（option）。

n　表示您必須給予一數目字。

1.2.2 語 法

所有可能的 DOS 指令格式，如下所述（圖 1 - 1）。

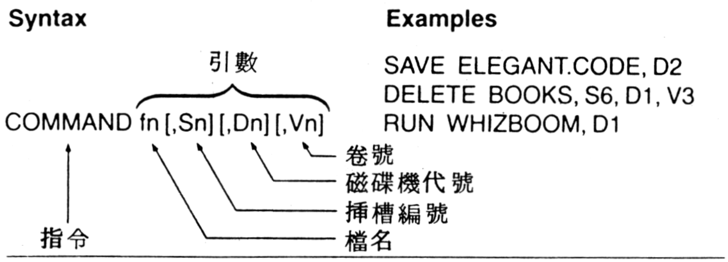


圖 1 - 1　DOS 指令的語法

圖 1-1 中，COMMAND 代表任何 DOS 指令（如 SAVE），fn，[，Sn]，[，Dn] 和 [，Vn] 是指令的引數，方括號內的引數表示選擇項，如果省略此項，則 DOS 採用預設值（default）。

1.2.2.1 預設值

當您沒有告訴 DOS 任何選擇性引數，DOS 會假設其值，稱做**預設值**（default）。預設值是，當您沒有正式給予引數值時，DOS

自動設定的假設值。

例如，您打入如下的指令：

```
CATALOG,D1,S6
RUN COPYA,D2（沒有指定槽號的指令）
```

第一個指令中您指定的擴充槽（slot）是第6槽，成爲預設槽（default slot）。當您在 RUN 指令中省略槽號（slot number）時，DOS自行以預設值第6槽爲槽號。

當您要更改預設值時，必須正式告訴DOS，可在指令中指定，爾後即用此值爲預設值。

1.2.2.2 數目

引數中小寫的 n 必須以數字加以取代，可以是十進位數目或 **16進位** 數目。

1.2.2.3 槽號選擇項(Slot Number Option:〔,Sn〕)

Sn 選擇項係指定您所要使用磁碟機之磁碟機控制界面卡（disk controller card）所在的擴充槽（slot）之號碼。將小寫 n 以 1 至 7 的任一數字取代之。

順便一提：在Apple II 和 Apple II Plus 電腦中，第 0 槽所插的是語言卡（Language Card）。

當您使用 Sn 選擇項時，您所指定的值即成爲預設槽號（default slot number），意即 DOS 繼續使用此槽號，直到您指定其他槽號爲止。

如果您沒有指定槽號，DOS找出最近使用的槽號。如果您一直

註：預設值是，當您沒有指定時，DOS付予的假設值。

註：16 進位數字系統（hexadecimal number system）是以 16 爲底的數目表示法，以數字 0 至 9 與 A 至 F 表示。

都沒有指定槽號，則 DOS 採用您啓動(boot) DOS所用的槽號。

1.2.2.4 磁碟機代號選擇項(Drive Number Option:〔,Dn〕)

Dn選擇項指定您所要使用的磁碟片所在的磁碟機代號，將小寫 n 以磁碟機代號 1 或 2 取代之。當您省略磁碟機代號時，DOS 採用最近使用的磁碟機，如果您徒未指定磁碟機代號，則 DOS 採用您啓動 DOS 的磁碟機。

1.2.2.5 卷號選擇項 (Volume Number Option:〔,Vn〕)

Vn 選擇項指定使用的磁碟片之卷號。雖然卷號選擇項可使用於大部份 DOS 指令，但是卻很少用到。當您使用 Vn 時，將小寫 n 以 0 至 254 之間任一數目加以取代，做爲卷號 (volume number)，此數目即成爲預設卷號 (default volume number)。

如果您使用 INIT 指令準備磁碟時省略卷號，則 DOS 付予預設値 254 給此磁碟片。在其他 DOS 指令中，如果您省略此選擇項或指定 V0 或只打V 沒有號碼，則 DOS 不管卷號。

1.2.2.6 16進位數字 (Hexadecimal Numbers)

在某些情況下，您可以將小寫 n 以 16進位數字取代之。機器語言程式師很喜歡 16 進位表示法，因爲要表示大數目時用 16 進位來表示，可以位數較短也比較方便。

在選擇項中，您無須使用 16 進位數字。但是如果您要採用的話，則以錢號 (dollar sign,$) 起頭然後接 16 進位數字 (hexadecimal digit)。例如，十進位數目 254 以 16 進位表示爲 $FE。

識別: 本書中，16 進位數字都以錢號 (dollar sign,$) 起頭。

註: To boot = to start up （啓動）

1.3 使用DOS指令

警告：DOS指令必須以大寫輸入。如果您使用Apple IIe，則必須按下 CAPS LOCK 鍵，使其處於大寫鎖定狀態。

1.3.1 CATALOG指令

CATALOG 指令是用來顯示整份檔案名錄，叫做**檔案目錄**（**catalog**）。檔案目錄列出此卷中所有檔案的檔名、大小和檔案型態（file type）。鎖住的檔案（locked file）列出時，在其檔案型態之前附有星號（asterisk *）。

如果檔案目錄太長，無法一次顯示於螢幕上，則必須按 RETURN 鍵，使其繼續列出其他檔案。

本指令的完整格式是：

CATALOG [, Sn] [, Dn]

[,Sn] 其中n是1至7之任一數字，指定您所使用的磁碟機之控制卡所在的擴充槽。如果省略此項，DOS採用預設槽號。

[,Dn] 其中n是1或2，指定所使用的磁碟機代號。如果省略此項，DOS使用預設磁碟機代號。

無卷號：如果您指定卷號（volume），則 DOS 不管此項。

註：檔案目錄（catalog）係列出某卷（volume）內所有檔案之名單，亦即是磁碟片內各檔案名稱的一覽表。

1.3.2 INIT指令

INIT 指令在磁碟片上建構磁軌（ track)和磁區（ sector ），並且將所有磁區填入 0 。同時，INIT 指令並將一份 BASIC 檔案和一份 DOS 拷貝存於磁碟片。以上的步驟建立一份**啓始化磁碟**（ initialized disk ）啓始化磁碟片上的 DOS 程式會讀入相同的記憶位址。

INIT 指令格式如下：

INIT fn [, Sn] [, Dn] [, Vn]

fn　　是用來做爲歡迎詞程式（ greeting program）或轉鑰式程式（ turnkey program ）的檔案之檔名。

[,Sn]　　其中 n 是 1 至 7 任一數字，指定您所使用的磁碟片所在磁碟機之控制卡所插的擴充槽。如果省略此項，DOS 採用預設槽號。

[,Dn]　　其中 n 是 1 或 2 ，指定存有歡迎詞程式或轉鑰式程式之磁碟片所在的磁碟機。如果省略此項，DOS 採用預設磁碟機代號。

[,Vn]　　其中 n 是由 0 至 254 任一數目，指定磁碟片所附的卷號。如果您指定爲 0 ，或只打 V，或省略此項，則DOS用 254做爲預設卷號。

註：啓始化的磁碟（ initialized disk)係準備提供Apple II電腦使用的磁碟片。

註：轉鑰程式（ turnkey program ）＝歡迎詞程式（ greeting program ）。

1.3.3 RENAME指令

RENAME指令修改檔案的檔名，將指定的 fn1 改名爲 fn2。指令格式如下：

RENAME fn1, fn2 [, Sn] [, Dn] [, Vn]

fn1,fn2 其中 fn1是已存在的檔案，而 fn2是新的檔名，必須唯一存在。如果 fn1不存在，畫面出現 FILE NOT FOUND 之錯誤訊息（error message）。此外，檔案必須先解鎖（unlock）。

[,Sn] 其中 n 是由 1 至 7 之任一數字，指定您所使用的磁碟片所在磁碟機之控制卡所插的擴充槽。如果省略此項，DOS 採用預設槽號。

[,Dn] 其中 n 是 1 或 2，指定存於要改名檔案之磁碟所在之磁碟機。如果省略此項，DOS 採用預設磁碟機代號。

[,Vn] 其中 n 是 0 至 254之任一數目，指定所使用的磁碟片之卷號。如果您指定爲 0，或只打入V，或省略此項，則 DOS 不管卷號。

警告：如果在執行RENAME之前 fn2已存在，則執行完後您已無法直接取得 fn2檔案原有的內容。

1.3.4 LOCK指令

LOCK指令用以鎖住檔案，即保護私人檔案，避免無意中被修改或刪除或改名。

LOCK fn [, Sn] [, Dn] [, Vn]

fn　　要上鎖（lock）的檔案之檔名。

[,Sn]　　其中n是由1至7的任一數字，指定您所使用的磁碟片所在磁碟機之控制卡所插的擴充槽。如果省略此項，DOS 採用預設槽號。

[,Dn]　　其中n是1或2，指定存有要上鎖的檔案之磁碟所在之磁碟機。如果省略此項，DOS採用預設磁碟機代號。

[,Vn]　　其中n是0至254之任一數目，指定所使用的磁碟之卷號。如果您指定爲0，或只打入V，或省略此項，則 DOS不管卷號。

當您試圖要修改或改名（rename）或刪除（delete）已鎖住的檔案（locked file），畫面出現 FILE LOCKED ，必須先利用UNLOCK指令爲此檔解鎖（unlock）。

在磁碟的檔案目錄（catalog）中，已鎖定的檔案其檔案型態左邊有一星號（asterisk *）。

順便一提：您無法將整個磁碟片鎖住，但是可在“可寫缺口”（write-enable notch）貼上防寫標籤（write protect tab），即可保護整片磁碟不至誤寫入資料。

1.3.5 UNLOCK指令

UNLOCK指令可爲一檔案解鎖（unlock），移去檔案的保護，即可從事刪除、改名或更改其內容。

UNLOCK fn [, Sn] [, Dn] [, Vn]

fn　　欲解鎖的檔案之檔名。

[,Sn]　　其中n是由1至7任一數字，指定您所要使用的磁碟片所在磁碟機之控制卡所插的擴充槽。如果省略此項，

DOS 採用預設槽號。

[,Dn]　其中n是1或2，指定存有要解鎖的檔案之磁碟片所在之磁碟機。如果省略此項，DOS採用預設磁碟機代號。

[,Vn]　其中n是0至254之任一數目，指定所使用的磁碟之卷號。如果您指定為0，或只打入V，或省略此項，則 DOS 不管卷號。

1.3.6　DELETE指令

DELETE 指令使您能將磁碟中的檔案除去。

DELETE fn [, Sn] [, Dn] [, Vn]

fn　是所要刪除的檔案之檔名，fn必須先解鎖。如果此檔不存在，畫面出現 FILE NOT FOUND。

[,Sn]　其中n是由1至7任一數字，指定您所要使用的磁碟片所在磁碟機之控制卡所插的擴充槽。如果省略此項，DOS 採用預設槽號。

[,Dn]　其中n是1或2，指定存有 fn 的磁碟片所在之磁碟機。如果省略此項，DOS採用預設磁碟機代號。

[,Vn]　其中n是0至254之任一數目，指定所使用磁碟之卷號。如果您指定為0，或只打入V，或者省略此項，則 DOS 不管卷號。

1.3.7　VERIFY指令

VERIFY 指令可以讓您檢查檔案寫入磁碟的資料是否正確無誤，檢查 DOS 是否能由此檔案讀取資料。

VERIFY fn [, Sn] [, Dn] [, Vn]

fn　指定您所要查證（verify）的檔案。任何型態的檔案皆能驗證，包括文字檔（text file）和機器碼檔案（binary file）。如果此檔並不存在，DOS 顯示 FILE NOT FOUND。

[,Sn]　其中 n 是由 1 至 7 任一數字，指定您所要使用的磁碟片所在磁碟機之控制卡所插的擴充槽。如果省略此項，DOS 採用預設槽號。

[,Dn]　其中 n 是 1 或 2，指定存有 fn 檔案的磁碟片所在之磁碟機。如果省略此項，DOS 採用預設磁碟機代號。

[,Vn]　其中 n 是 0 至 254 之任一數目，指定所使用磁碟片之卷號。如果您指定爲 0，或只打入 V，或者省略此項，則 DOS 不管卷號。

如果檔案能完成驗證，則可相信磁碟中的資訊保存無誤，可隨時擷取這些資料。

在驗證檔案時，DOS 只是將檔案由磁碟讀入檔案保留區（file buffer），記憶體中此保留區係目前沒有使用的一塊區域。因此，這樣做不會摧毁已在記憶體中的程式。如果 DOS 能成功地讀取檔案，則直接出現提示字元（prompt character）。如果 DOS 發現無法讀取檔案，表示檔案被破壞或寫入不正確，畫面會出現 I/O ERROR.

沈默是金：如果 DOS 能讀檔案，則 DOS 不會顯示任何訊息。

註：有關檔案保留區，參考第 5 章。

1.3.8 RUN指令

RUN指令可執行存於磁碟中的 Applesoft 或 Integer BASIC 程式。

RUN fn [, Sn] [, Dn] [, Vn]

fn　　指示 DOS 執行的程式，其檔案型態必須是A或 I 。如果程式不在磁碟中，則畫面出現 FILE NOT FOUND 。如果檔案型態不是A也不是 I ，您會看到 FILE TYPE MISMATCH 的字樣。如果 DOS 無法切換至程式所屬的語言系統，畫面會出現 LANGUAGE NOT AVAILABLE 。

[,Sn]　　其中 n 是由 1 至 7 之任一數字，指定您所使用的磁碟片所在磁碟機之控制卡所插的擴充槽。如果省略此項，DOS 採用預設槽號。

[,Dn]　　其中 n 是 1 或 2 ，指定存於要執行程式之磁碟所在之磁碟機。如果省略此項，DOS採用預設磁碟機代號。

[,Vn]　　其中 n 是 0 至 254 之任一數目，指定所使用的磁碟片之卷號。如果您指定爲 0 ，或只打入 V，或省略此項，則 DOS 不管卷號。

當 DOS 看到 RUN 指令，會去找指定的程式，將其帶入記憶體，並啓動它。將程式帶入記憶體之前，DOS先檢查程式的檔案型態。如果檔案型態是A而且 Applesoft BASIC 沒有處於使用型態，DOS會自動切換成 Applesoft（如果可能的話）。如果檔案型態是 I 而且 Integer BASIC 不處於使用狀態，DOS會切換成 Integer（如果可能的話）。

註: 有關 BASIC 程式檔，參考 2.2 節。

因爲 RUN 指令會自動將程式載入記憶體，因此在執行程式之前不需要使用LOAD指令。

順便一提： 只要程式進入記憶體之後，利用 RUN 指令無須加上檔案名稱，即可再執行此程式一次。沒有檔名的 RUN是 BASIC 指述（statement）。

1.3.9 LOAD指令

LOAD指令將磁碟程式抄入記憶體。

LOAD fn [, Sn] [, Dn] [, Vn]

fn 指定DOS 要載入的程式，其檔案型態必須是A或I 。如果程式不在磁碟中，畫面出現 FILE NOT FOUND 。如果檔案型態不是A或I ，則您會看到 FILE TYPE MISMATCH。如果DOS 無法切換到程式所屬的語言，您會看到 LANGUAGE NOT 的字樣。

[,Sn] 其中n是由1至7之任一數字，指定您所使用的磁碟片所在磁碟機之控制卡所插的擴充槽。如果省略此項，DOS 採用預設槽號。

[,Dn] 其中n是1或2，指定存有要執行程式之磁碟所在之磁碟機。如果省略此項，DOS採用預設磁碟機代號。

[,Vn] 其中n是0至254之任一數目，指定所使用的磁碟片之卷號。如果您指定爲0，或只打入V，或省略此項，則 DOS 不管卷號。

將程式帶入記憶體之前，DOS檢查程式的檔案型態，並試圖切換到適當的 BASIC 。當您利用LOAD指令將檔案內容載入記憶體後，磁碟上的檔案內容保持不變。

在您使用 RUN 指令之前，無須先將程式載入記憶體，RUN會

自動將其載入的。

順便一提：當新程式載入記憶體後，上一個程式從Apple的記憶體消失。如果您不希望毀掉目前記憶體中的程式，則必須利用SAVE指令將其保存起來。

1.3.10 SAVE指令

SAVE指令將目前記憶體中的BASIC 程式儲存到磁碟中。

SAVE fn [, Sn] [. Dn] [, Vn]

fn 指定程式的檔案名稱。如果此名稱的檔案已存在，則此檔案必須先解鎖（unlock），而且必須與您要保存的程式具有相同的檔案型態（A或I）。

[,Sn] 其中n是由1至7之任一數字，指定您所使用的磁碟片所在磁碟機之控制卡所插的擴充槽。如果省略此項，DOS 採用預設槽號。

[,Dn] 其中n是1或2，指定存有要執行程式之磁碟所在之磁碟機。如果省略此項，DOS採用預設磁碟機代號。

[,Vn] 其中n是0至254之任一數目，指定所使用的磁碟片的卷號。如果您指定此項爲0，或只打入V，或省略此項，則DOS 不管卷號。

當提示字元（prompt character）是]時，DOS將程式寫入磁碟所定的檔案型態是A。當提示字元是>時，DOS寫入的檔案其型態是I。除了爲您處理檔案型態之外，DOS自動查出磁碟位址以及要保存的檔案之長度。

註：有關 BASIC 程式檔，請參考2.2節。

警告: 當您要保存檔案時，如果出現 I/O ERROR ，可能是磁碟損壞 (bad)或是沒有啓始化步驟 (initialized) ，或者是您所指定的磁碟機沒有放磁碟片，也許是磁碟機蓋子沒有蓋上。放入一片新的，已啓始的磁碟片，關上磁碟機蓋子，再使用 SAVE 指令。

---

2 DOS的程式工具

2.1 使用DOS指令的方法 21
2.1.1 立即執行模式 22
2.1.2 延遲執行模式 22
2.1.3 指令檔 22
2.2 BASIC 程式檔 23
2.2.1 利用FP和INT指令切換語言 25
2.2.1.1 FP指令 26
2.2.1.2 INT指令 26
2.2.2 由監督程式回到BASIC 27
2.3 程式中使用DOS 28
2.3.1 範例 29
2.3.2 注意事項 30
2.4 利用MON和NOMON偵錯 32
2.4.1 MON指令 33
2.4.2 NOMON指令 33
2.5 利用Applesoft來攔截錯誤 34
2.5.1 範例 35
2.5.2 找出檔名中的隱藏字元 37
2.6 在程式中利用其他週邊設備 38
2.6.1 利用PR#和IN#來啓動程式 40
2.6.2 PR#指令 40
2.6.3 IN#指令 41
2.7 RENUMBER程式 42

2.7.1 RENUMBER 指令 43
2.7.1.1 重編號指令的語法 44
2.7.1.2 內部"列號叫用" 45
2.7.1.3 駐留區 46
2.7.1.4 範例 46
2.7.2 RENUMBER的錯誤訊息 48
2.8 摘要 50
2.8.1 指令 50
2.8.2 程式 51

第2章
DOS的程式工具

本章討論如何用鍵盤（ keyboard ）或從程式中來使用 DOS 的指令，要點如下：

- 如何切換兩種不同的 BASIC 語言系統，以及何時使用。
- 如何監督指令的執行。
- 如何在程式中找出錯誤，在檔案名稱中找出隱藏字元（ hidden character ）。
- 如何讓您的程式與擴充槽上的週邊裝置溝通。
- 如何將您的程式重新編排列號。

本章和以下各章節描述進一步使用DOS 的方法，最好能先熟悉 BASIC 的程式。如果您已做完“ Applesoft Tutorial ”中所有的例子，則您已具備基本的程式觀念，有能力使用本書。

2.1 使用 DOS指令的方法

您可以從鍵盤直接下達 DOS 的指令，也可以將其置於BASIC 程式中，或者將指令放入指令檔內。

2.1.1 立即執行模式

您可以從鍵盤上直接下達 DOS 指令，稱爲立即執行（Immediate execution）模式。大部分 DOS 指令都可以利用這種方式執行，但有少部分指令不行。

當您從鍵盤打入指令時， DOS 先檢視指令，然後從有效的 DOS 指令表中找出此指令。如果找不到，則將其傳回給BASIC。當您輸入 DOS 指令時，有時會有錯誤訊息出現，這時您應檢查是否使用適當的 DOS 指令格式，然後再重新輸入。

2.1.2 延遲執行模式

您可以在 BASIC 程式中，使用大部分 DOS 指令，此方法稱**延遲執行**(deferred execution)模式，因爲程式中的指令必須等到程式 run時，才會執行的。例如，如果程式含有 DOS 的OPEN指令時，程式即可開啓（ open）一磁碟檔以便儲存資料。因此，此 DOS 指令一直暫緩動作直到您用 RUN 指令執行此程式才眞正有作用。

2.1.3 指令檔

可以執行其他程式的檔案稱做**指令檔 (command file)**。因爲EXEC 指令用來執行指令檔，所以指令檔又稱做EXEC檔（EXEC file）。指令檔含有DOS 指令和BASIC 的指述（statement

---

註: 立即執行：指令由鍵盤輸入。

註: 錯誤訊息（ error message ）請參考附錄C。

註: 延遲執行模式的(deferred execution)指令直到程式執行（run）時才眞正有動作。

2.2 BASIC程式檔

Apple II電腦提供兩種 BASIC（Beginner's All-purpose Symbolic Instruction Code，初學者萬用符號指令碼）程式語言：Applesoft BASIC和 Integer BASIC。Integer BASIC（整數培基）是 BASIC 語言的最早版本，僅能處理**整數**（Integer）。Applesoft 除了能處理整數之外，也可處理具有小數點的數目，亦即是具有**浮點運算**（floating point）的能力，這是兩者最大的不同點，欲深入瞭解，可參考“Applesoft BASIC Programmer's Reference Manual”。

每一台Apple II電腦都將這兩種 BASIC 中的一種燒在**唯讀記憶體**(Read-Only Memory,ROM)中。這類常駐語言（resident language）只要打開您的電腦即可使用。在標準的Apple II中，Integer BASIC 是其常駐語言（resident language）;Apple II Plus 和Apple IIe則採用 Applesoft BASIC 爲其常駐語言。

使用那一種語言與您期望電腦做的事情有關。您能利用的語言則與您的電腦之記憶容量以及其硬體或韌體（firmware）有關。例如，如果您的Apple II電腦擁有 48K 或更少的**隨讀記憶體** (Random Access Memory,RAM)，則您只能擁有一種語言，除非您加入界面卡（card）擴充成更多的記憶體。

註: EXEC檔案＝指令檔（command file）。

註: Applesoft BASIC Programmer's Manual 有一附錄扼要比較 Integer BASIC 和 Applesoft BASIC 之間的差異。

註: 欲加入不同的語言，可加裝 Applesoft 卡（card）、整數培基卡（Integer Card）或者語言卡（Language Card，或稱做 16 K RAM Card）。

很多Apple II電腦都有足夠的記憶體同時容納這兩種語言。至於Apple IIe,則無須加上任何硬體即可容下這兩種語言。

當您在64K以上記憶體的Apple II上啓動 DOS 時，DOS 會從磁碟載入（ load）“非常駐的”（ nonresident ）BASIC 。DOS 會向您報告，到底將那一個 BASIC 載入記憶體。如果電腦係常駐的 Applesoft BASIC ，您會看到螢光幕顯示 LOADING INTEGER BASIC INTO MEMORY ；如果您的Apple係常駐的 Integer BASIC ，則您會看到 LOADING APPLESOFT BASIC INTO MEMORY 的字樣。如果您的Apple II記憶體小於48K或者啓動磁碟（ startup disk ）沒有另一種 BASIC 時，則不會出現 LOADING …的訊息。

要判別您正在使用那一個 BASIC ，只要看一看提示字元（ prompt character ）即可分曉：

] 表示Applesoft BASIC 。

> 表示Integer BASIC 。

當您想從某一個語言轉到另一個時，可利用指令 FP 或 INT 。當您處於 Applesoft BASIC的狀態時，意即提示字元是] 時，打入 INT 指令，即可轉成 Integer BASIC ，此時螢幕上的指示字元是>。

警告: 使用RUN 或LOAD指令時，會自動執行或載入屬於BASIC原始程式的檔案。

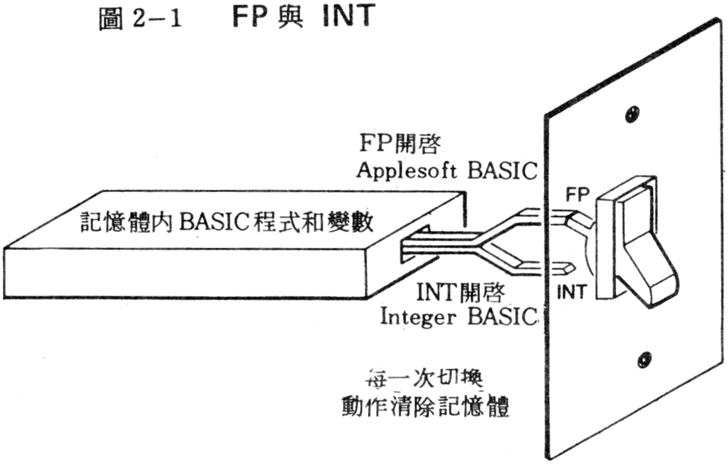
在 Apple II Plus 和Apple IIe 電腦中，當 Integer BASIC是“非常駐語言（ non-resident language ），意即它不在記憶體時，會出現 LANGUAGE NOT AVAILABLE 的字樣。

在標準的 Apple II 電腦中， Integer BASIC 係常駐語言。當Applesoft BASIC 不在記憶體中時， DOS 會找出Applesoft 程式，並加以執行。如果Applesoft BASIC 不在磁碟中，您會看到 LANGUAGE NOT AVAILABLE. 。

2.2.1　利用FP和INT指令切換語言

當兩種 BASIC 語言都在記憶體時，可利用 FP 和 INT 來切換這兩種語言系統。當您利用 FP 或 INT 指令回到原先您就在使用中語言時，則系統和記憶體相當於重新設定（ reset ）恢復成啓始狀態。兩指令皆適用於立即執行（ immediate execution ）和**延遲執行**（deferred execution）兩種方式。

圖2-1　FP與 INT



註:APPLESOFT程式將存有Applesoft BASIC 的磁碟檔案 FPBASIC 載入語言卡（Language Card ）。

警告: FP 和 INT 指令會重設記憶體中的指標（ pointer ），因此您的程式和變數會從主記憶體中消失。如果您要切換語言，必須在打入新程式之前使用這二個指令。

2.2.1.1 FP指令

FP 指令將系統切換到Applesoft BASIC，FP 是代表浮點(floating point)。如果您正在使用 Integer BASIC ，然後又希望寫一程式來平衡您的支票簿，則您必須切換到Applesoft BASIC，才能處理實數。

此指令的格式如下：

FP [, Sn] [, Dn] [, Vn]

[,Sn] n 是 1 至 7 之號碼，指明所使用的磁碟機之界面卡所在的擴充槽號碼。如果沒有指定，則DOS用預設的槽號。

[,Dn] n 是 1 或 2 ，指明要使用的磁碟機（ drive ）的號碼。如果省略不提，則 DOS 採用預設的磁碟機號碼。

[,Vn] n 是 0 至 254 的數字，指定所要使用的磁碟片之卷號（volume number ）。如果只寫V即 0 號，或者省略此選擇項（ option ），則 DOS 不管卷號（ volume specification ）。

警告: 千萬不要用RUN APPLESOFT來變換語言。如果您這麼做，則 DOS 檢查檔案型態（file type），並調整 IntegerBASIC 的記憶指標（memory pointer），而不是Applesoft BASIC 的。參考第6 章，記憶體的使用與進入點（entry point）。

2.2.1.2 INT指令

INT 指令用來切換到 Integer BASIC語言卡， INT 代表整數（ integer ）。完整的指令格式如下：

INT

INT 指令沒有附帶任何引數。

2.2.2　由監督程式回到BASIC（Returning to BASIC From the Monitor Program）

有時畫面會出現星號 *，即是**監督程式(monitor program)**的提示字元（prompting character）。也許是因爲您正在執行的程式有錯，此錯誤的程式跳入監督程式（Monitor）中。

有三種方法可以從監督程式（Monitor）回到 **BASIC**。其中有兩種方法可以恢復正常工作；第三種方法則會毀壞記憶體中的資料，因此您必須重新開始。

1. 重新啓動 **DOS** 並使您的程式回到 **BASIC**，同時保留變數（variable）完整，則打入

   3D0G　　　　　　　（注意，那是零…）

   這不是 DOG：3D0 是 **DOS** 在記憶體中的位址，而 G 表示執行（go）。

2. 如果要回到 **BASIC** 且不更動您的程式和變數（variable），請打入 CONTROL-C 並按下 RETURN 鍵。

警告：CONTOL-C 只用來重返您尚未進入監督程式（monitor）之前所使用的 **BASIC**。如果在 CONTROL-C 之前，**DOS** 就已不在，雖然您想回到 **BASIC**，但是依舊沒有作用。

---

3. 以下的指令也可以回到 **BASIC**：

   CONTROL - B 或 3D3G

警告：CONTROL-B 和 3D3G 會將您的程式和變數從記憶體中消除掉。

註：有關監督程式（monitor），請參考"**Apple II** Refernce Manual"。

2.3 程式中使用 DOS

在 BASIC 程式中使用 DOS 指令，可以擴展您的程式的能力。例如，您可以用延遲執行模式（deferred-execution）的 DOS 指令，自動顯示磁碟目錄（disk's catalog），將檔案中某些資料錄（record）的儲備副本（back-up copy）保存起來，或者保留未完的遊戲之目前狀態以便稍後繼續。

要在程式中使用 DOS 指令，可利用 BASIC 的 PRINT 指述（statement），跟隨一字串。在此種情況之下，此字串包含 DOS 的指令。要指明您是要輸出"DOS 指令"而不是"文字"，必須在字串的第一個字母打入 CONTROL-D，其一般格式是:

PRINT "CONTROL-D DOS指令"

有兩種方法可以將 CONTROL-D 輸入您的程式，其中之一法 CONTROL-D 會出現在 PRINT 指述，另一法則不會。

如欲採用隱藏的 CONTROL-D，則直接在雙引號（quote mark "）之後打入 CONTROL-D，做為此字串的起始。

10 PRINT "CATALOG" （CONTROL-D 藏在"與C之間）

雖然您看不見，但是 CONTROL-D 確實存在。這個方法無論是 Integer BASIC 或 Applesoft BASIC 的程式指述皆適用。

註：如果您省略 CONTROL-D，DOS 將此字串視爲 BASIC 指述（statement）的一部分，將其傳給 BASIC 的指令解譯程式（command interpreter）處理，而不視爲 DOS 指令。

警告：利用→鍵本來可以複製 BASIC 指述，但是卻無法複製隱藏的控制字元（control character）。事實上，→鍵會刪除掉字元。

只有在 Applesoft **BASIC**，您才能"看到" [CONTROL]-[D]。此方法是利用CHR$ **函數**(function)。CHR$函數利用一數字引數 (arguement)，求出其對應的 **ASCII** 字元。

[CONTROL]-[D] 的 **ASCII 碼** (American Standard Code for Information Interchange ，美國標準資訊交換碼) 是 4 。當您將數值 4 輸入 CHR$ ，即可得到 [CONTROL]-[D]。

只要在您的程式開頭之處，將 CHR$ (4)的值指定給某一字串變數 (string variable)，即可利用 **BASIC** 指述在各 **DOS** 指令之前，輸出 (print) 此字串變數。

在 **PRINT** 指述中，利用 CHR$函數 (function) 產生 [CONTROL]-[D] 的例子如下：

```
5 REM VALID ONLY WITH APPLESOFT
10 D$=CHR$ (4)
20 PRINT D$;"CATALOG"
```

在本書中都以 D $ 表示存有[CONTROL]-[D]的字串變數，此變數的名稱可幫助我們記住它所含的控制字元 (control character) 是 [CONTROL]-[D]。

分號 (semicolon ;) 可省略： 如果您的程式有很多 **PRINT** 指述要利用 **DOS** 指令，省略分號可以節省輸入時間和記憶空間。如果沒有分號，則列 20 的指述如下：

```
20 PRINT D$"CATALOG"
```

2.3.1　範　例

以下的 Applesoft **HELLO** 程式顯示一段訊息和磁碟的目錄 (catalog) 於螢幕上。請打入**NEW**以清除記憶體中原有的程式，並輸

註: 函數 (function) 可求出一數值，供爾後計算之用。

入此程式。

```
5 REM APPLESOFT HELLO
10 D$ = CHR$ (4) : REM CONTROL-D
                              ( D$是 CONTROL - D )
20 PRINT "DOS TEST PROGRAMS"
30 PRINT " 26 NOVEMBER 1985"
                              (( 顯示標題和日期 ))
40 PRINT D$;"CATALOG"         ( 然後列出檔案目錄 )
50 END
```

現在打入 **RUN**，並看看執行的結果。

如果利用 Integer **BASIC** 來寫，則程式如下：

```
5 REM INTEGER HELLO
10 D$ = "" : REM CONTROL-D
                              ( 兩個引號" 之間藏有
                              CONTROL - D )
20 PRINT "DOS TEST PROGRAMS"
30 PRINT " 26 NOVEMBER 1985"
                              ( 顯示標題與日期 )
40 PRINT D$;"CATALOG"         ( 然後列出檔案目錄 )
50 END
```

注意列號 10 的指述內容，那是兩者之間唯一不同的地方。如果您採用這種方式來設定 D $ 變數 (variable) 爲 CONTROL - D ，最好加入註解， REM CONTROL-D ，來說明隱藏的字元。

2.3.2 注意事項

在 Applesoft **BASIC** 程式中使用 **DOS** 指令，必須注意三件事情。

註: 從現在起，本書大部分的例子皆採用 Applesoft **BASIC** 寫法。

第一，每個 PRINT 指述只能有一個 DOS 指令。

第二，在列印D $之前必須有歸位字元（ carriage return ）。如果 DOS 指令之前的指述以分號（ semicolon ; ）、定位字元（TAB）或逗號（ comma ,）結束，則此指令無法傳給 DOS 指令解譯程式（ command interpreter ）。

因此，如果 DOS 指令不是從新的一列開始，則指令不會被執行而是印出指令名稱以下的例子中，CATALOG這個字會印在螢幕上，而不是執行此指令：

```
30 D$ = CHR$ (4) : REM CONTROL-D
                                  （D$是 CONTROL - D ）
40 PRINT "AUTUMN ";               （分號制止RETURN產生）
50 PRINT D$;"CATALOG"             （無法產生檔案目錄）
```

此程式不會列出磁碟的目錄，而是印出 AUTUMN CATALOG 等字樣。

警告：當您的程式出乎意外地印出 DOS 的指令名稱，而不執行此指令，則請趕緊找出是那一條指述以分號結束之。

如欲保證在 CONTROL-D 之前有印出歸位字元，則將變數D $的值設定爲歸位字元（ ASCII 是 13 ）和 CONTROL-D （ ASCII 是 4 ）兩個字元：

```
D$=CHR$(13)+CHR$(4)
```

當D $以此種方式宣告時，Applesoft 指述 PRINT D$;"CATALOG" 會在 CONTROL-D 之前印出" carriage return "字元（

註：如果 PRINT 指述以分號結束，則在印出字串（ string ）、變數（ variable ）或文字（ literal）之後不會自動加上歸位字元。

即游標跳回第一格），如此可保證 DOS 指令必然從新的一列開始。

警告：利用這種方式來宣告D $時，如果您將其寫入（write）或增添（append）於文字檔，將會造成問題。

第三點注意事項是，有些 DOS 的指令僅能用於程式。僅適用於延遲執行(deferred execution)模式的 DOS 指令：APPEND（增添）、OPEN（開啓）、POSITION（定位）、READ(讀出）、WRITE（寫入）。

2.4 利用 MON和 NOMON 偵錯

程式的錯誤常稱爲“**臭蟲**”(bug)；修改程式使其能正確執行的整個執行過程，謂之**偵錯**(debugging，抓蟲）要爲程式偵錯，您可能需要追踪磁碟與電腦之間的資訊交換過程，監督這些行動，程式設計者比較容易找出問題的癥結。

欲監視整個執行過程可使用“監督指令”MON。利用MON，您可以監視“開啓指令”OPEN如何管制磁碟檔案的操作，清楚地獲知 PRINT 指述輸入磁碟的結果，或者READ指令從磁碟讀入的資料。

如果您只想看程式眞正執行的結果，希望程式能夠快速執行，這時可利用“解除監督指令”NOMON消除這些多餘的訊息。

最佳拍檔：偵錯時如欲知道程式的指述編號（statement number），可將 DOS 的 MON、NOMON指令與 BASIC 的 TRACE（追踪）、NOTRACE（解除追踪）指述加以混合使用。詳細用法可參考Applesoft BASIC Programmer's Reference Manual。

當使用追踪指令TRACE時，要確保變數 D $ 存有 car-

riage return和 [CONTROL]-[D] 兩個控制字元，亦即是必須設定D $的值爲，D$ = CHR$(13) + CHR$(4)。

2.4.1　MON指令

平常我們並不需要監視所有的磁碟指令或電腦與磁碟之間傳送的資訊。如果要監督這些動作，就必須利用“監督指令”MON。MON 指令的語法如下：

MON [C] [,I] [,O]

C　顯示所有的磁碟指令（disk command）。

I　顯示輸入結果（input），即由磁碟輸入電腦的資訊。

O　顯示輸出結果（output），即由電腦輸出磁碟的資訊。

引數C、I、O分別代表指令（command）、輸入（Input）和輸出（Output）。根據您所希望監督指令MON 提供的資訊，可以任意選出您所要的引數並隨意排列之。而逗點則可有可無，例如 MON CIO。

警告：至少要選用一個引數，否則此MON 指令沒有作用。

除非您又下達NOMON指令，否則MON一直保持有效。另外，FP 指令、INT 指令、用啓動磁碟（startup disk 或 boot disk）重新啓動 DOS、使用3D0G指令或者用 3D3G，都會使MON失效。

2.4.2　NOMON指令

“解除監督指令”用來消除磁碟指令的顯示以及抑制電腦與磁

碟之間的資訊顯示出來。NOMON指令的語法如下：

NOMON[C] [,I] [,O]

C 不顯示所有磁碟指令。

I 不顯示從磁碟輸入電腦的資訊。

O 不顯示從電腦輸出磁碟的資訊。

引數（ argument ）C、I、O分別代表指令（Command ）、輸入（ Input ）和輸出（ Output ）。這些引數得以任意組合與任何順序出現，完全依您希望消除那些資訊來取捨，逗號則可有可無。例如， MONICO.

警告：至少要選用一個引數，否則此NOMON 指令沒有作用。

利用以下的指述來使用NOMON指令時，即使處於監視狀態，也不會顯示此列指令的名稱。

```
10 PRINT D$; "NOMON C,I,O": VTAB PEEK(37):
     CALL -868
```

其中D $存有CONTROL-D; VTAB PEEK (37) 將游標（ cursor ）移至NOMON C,I,O這一列的開頭；CALL-868則消除此列。

2.5 利用 Applesoft 來攔截錯誤

當 DOS 或Applesoft 檢查到磁碟的使用有錯誤時，其平常的處理方式是立即停止有錯的程式，印出**錯誤訊息**（error message ），並將錯誤編號（ code number for error）存入記憶位址222。

註：錯誤訊息之解說請參考附錄C所示。

您可以設計一個Applesoft 的程式，防止 DOS 和Applesoft 有錯時，即被強制性的中止執行。此程式會檢查記憶體，並傳回錯誤編號（error number）以便檢查。根據這個編號，您的程式可跳至某一列來處理此類的錯誤。欲建立處理錯誤（error-handling）程式，可運用ONERR GOTO指述和PEEK函數的程式技巧。

- ONERR GOTO指述係指定一個指述列號（statement number），以便程式有錯時轉變程式的執行次序（transfer control）。您的處理錯誤副程式（error-handling routine）即可從ONERR GOTO指述所指定的指述列再開始執行。
- PEEK函數用以檢查記憶體中某一特定位址所存的內容，位址以十進位數值來表示。欲檢查錯誤編號，可檢查記憶位址 222；利用 PRINT PEEK (222) 指述可從螢幕上看到錯誤編號。您也可以將PEEK(222)的錯誤編號存入數值變數（numeric variable）。如欲知道造成錯誤的指述列號（statement number），利用 PEEK (218) + PEEK (219) * 256 即可得知。在上述的運算式（expression）之前加上PRINT，即可將指述列號顯示於螢幕上。

2.5.1 範 例

ONERR GOTO這個程式告訴您如何利用ONERR GOTO指述和PEEK函數來挽救DOS的錯誤。此程式傳回錯誤編號，並且報告造成錯誤的指述列號。

ONERR DEMO可以將一檔案改名，無論此檔案是否有鎖住

註：PEEK用以求一變數的值，PRINT PEEK(位址值)將變數顯示於螢幕上。

（ lock ）。當ONERR DEMO檢查到錯誤時，利用 ONERR GOTO指述，跳至錯誤處理副程式執行。錯誤處理副程式 使您有機會，在爲此檔案再改名之前，先爲此檔案解鎖（ unlock ），而後再印出錯誤編號（ error number ）和錯誤的列號（ line number)。

```
5 REM ONERR DEMO
10 D$ = CHR$ (4)                (D$是 CONTROL - D )
15 PRINT D$;"MON C"             (觀察磁碟指令)
20 ONERR GOTO 100               (列100 處理錯誤)
30 INPUT "FILE TO RENAME? ";F$
                                (檔名讀入F$
40 INPUT "NEW NAME? ";N$
                                (新檔名，讀入N$ )
50 PRINT D$;"RENAME ";F$;", ";N$
                                使用RENAME指令)
60 END                          (程式結束，無誤)
```

列 30 將欲改名之檔案名稱讀入並存於F $ ；列 40 則讀入新檔名於N $ ；列 50 是改名指令RENAME。當沒有錯誤發生的情況，此程式爲指定的檔案改名，然後結束。當有錯誤發生時，列 20 的ONERR指述（ statement ）將執行程序轉入列 100 來處理錯誤。

```
100 PRINT: IF PEEK (222) <> 10 THEN 200
```

列 100 檢查位址 222 的值，如果位址 222 的值是 10（ FILE LOCKED檔案鎖住的錯誤編號），則程式執行轉入列 110 ，顯示一段“已鎖住”訊息，要求打入Y或N再度確認。如果位址 222的內含值不是 10 ，則表示這是其他的錯誤，程式轉入列 200 執行。

```
110 INPUT "FILE IS LOCKED, RENAME ANYWAY? (Y/N)
    ";Y$                （檔案已上鎖，需要改名否？）
120 IF Y$ <> "Y" THEN 160
                         （不要改名）
130 PRINT D$;"UNLOCK ";F$
                         （要改名，為檔案解鎖）
140 PRINT D$;"RENAME ";F$;", ";N$
                         （改檔名）
150 PRINT D$;"LOCK ";N$  （再度鎖檔）
160 PRINT D$;"NOMON C": END
                         （抑制指令的顯示）
```

列 130 為檔案解鎖（ unlock），列 140 進行改名動作，列 150 再度為新命名的檔案加鎖（ lock ）。

```
200 PRINT "ERROR #";PEEK (222);" DETECTED"
210 PRINT "AT LINE ";PEEK (218) + PEEK (219) * 256
```

當程式的錯誤不是檔案鎖住（ **FILE LOCKED** ）時，列 200 和列 210 顯示錯誤編號和錯誤發生的指述列號。

由**SAMPLE PROGRAMS**磁碟片將程式**ONERR DEMO**載入執行，並試著為鎖住的檔案改名，請打入

```
RUN ONERR DEMO
```

2.5.2　找出檔名中的隱藏字元

您可以利用 Applesoft **BASIC** 來找出隱藏字元（ hidden character ）。當檔名（ file name ）含有控制字元（ control char-

---

註：改名指令**RENAME**無法為鎖住的檔案改名。當您的程式要為鎖住的檔案改名時， **DOS** 會傳回 FILE LOCKED 檔案鎖住的訊息和錯誤編號（ error-code number）10 。

註：詳細的 **DOS** 錯誤編號及其解說，請參考附錄 C 。

acter)時，此字元無法印出，所以看不到，但是使用檔案或刪除檔案時，還是必須將全名打入。

如果您無意中將控制字元引入檔案名稱之中，可以利用Applesoft程式找出任何隱藏字元，但是以下字元例外：CONTROL-M（歸位字元 carriage return）、ESC、CONTROL-H（←）、CONTROL-U（→）。

```
10 DATA 201, 141, 240, 21, 201, 136
20 DATA 240, 17, 201, 128, 144, 13
30 DATA 201, 160, 176, 9, 72, 132
40 DATA 53, 56, 233, 64, 76, 249
50 DATA 253, 76, 240, 253
60 FOR I = 768 TO 768 + 27
70 READ V : POKE I,V : NEXT I
80 POKE 54,0 : POKE 55,3
90 CALL 1002
```

打入這個程式，保存起來，並加以執行。當您下達目錄指令CATALOG，檔名中的控制字元會以閃爍（ blink ）的方式顯示出來。

如欲使Apple II Plus電腦恢復正常的顯示方式，只要打入指令 PR# 0 印可。

在 Apple IIe機器中，如果沒有開啓 **80格文字卡** (80–Column Text Card) 功能，打入PR#0 會回到標準的 40 格顯示。如果您已開啓 80 格文字卡，打入指令PR#3 會回到 80格顯示，或打入PR#3 和 ESC 、CONTROL-Q則回到 40 格顯示。

警告: 在Apple IIe機器中，當80格文字卡開啓時，千萬不要使用PR#0 指令，否則會得到難以預料的結果。

2.6 在程式中利用其他週邊設備

本節中，您將學習如何使程式與週邊設備（ peripheral dev-

ice ）溝通，例如：印表機（ printer）或磁碟機（ disk drive ）。這些裝置係透過插於擴充槽（ expansion slot ）上的界面卡（card ）與電腦相連接。

您的Apple II電腦通常將文字送到顯示螢幕；顯示器即當做標準輸出設備（ standard output device ）。同時從鍵盤（ keyboard ）讀入字元；鍵盤即是標準輸入設備（ standard input device ）。PR#（輸出）和 IN#（輸入）指令讓您得以使用其他設備。

警告：同一時間不得有一個以上的週邊設備同時處於動作狀態（ active state ）。

在 DOS 狀態之下：PR#和IN# 可採“立即執行”模式，自鍵盤下達指令執行的，請參考 BASIC 的手册。但是，如果PR#和IN#指令係在程式中下達，則必須利用 PRINT 指述，以 CONTROL - D 的技巧，指明它們是 DOS 指令（延遲執行模式）例如，

```
20 PRINT D$; "PR# 1"
30 PRINT D$; "IN# 2"
```

使用延遲執行（deferred execution ）的方式來執行指令PR#和IN#時，如果省略 CONTROL - D ，則此指令不再是DOS指令而是 BASIC 指述。則 DOS 不會與此設備連接，即無法恰當地印出或讀入字元。

如欲使 DOS 的輸入（ input）和輸出（output ）恢復到標準設備（ standard device ），可利用 BASIC 指述：

```
CALL 1002
```

註：欲知詳情，請參考6.4.1節。

2.6.1 利用 PR# 和 IN# 來啓動程式

使用輸出指令PR#和輸入指令 IN#的主要目的是指定輸入(input)和輸出(output)裝置。您可以利用此指令，將指令中指定的磁碟機(連接於指定的擴充槽上)內磁碟的程式予以啓動。

當您使用PR#和 IN#指令， DOS 會執行該擴充槽界面卡(card)所裝設的 ROM 程式。如果該 ROM 程式係磁碟控制卡(disk controller card)所有，則磁碟中控制器(disk controller)會自動由磁碟中讀入資訊。"從磁碟中讀入資訊"意謂著載入啓始程式(boot program)並加以執行，即啓動Apple II。

2.6.2 PR# 指令

輸出指令PR#指明電腦的某一擴充槽(slot)，並將連於此槽的設備作爲輸出字元的"目的地"。

輸出指令PR#也可以指定您的磁碟控制卡(disk controller card)，來執行磁碟上的程式。PR# 指令其語法如下：

PR# n

其中 n 是從 1 至 7 的數字。 DOS 將字元送到連於指定槽(slot)的設備上。數字記號 # 是指令的一部份，必須要打入，不可省略。

例如，當您的電腦有一印表機控制卡(printer controller card)插於第 1 槽，而且您希望您的程式將輸出(output)指向印表機(printer)時，即可利用如下的指令：

```
10 D$=CHR$(4)
20 PRINT D$; "PR# 1"
```

如果希望將輸出設備恢復到螢幕，在標準的Apple II或Apple II Plus電腦，則利用指令PR＃0。

在Apple IIe電腦，當您未啓用80格文字卡(80—column Text Card)時，利用PR＃0指令可回復成40格螢幕。如果80格文字卡正使用中，利用PR＃3指令可恢復成80格顯示，或者利用PR＃3指令，再打[ESC]和[CONTROL]-[Q]即可成爲40格顯示。

順便一提：PR #0是特殊情況，宣告DOS將輸出設備設定爲螢幕。在標準的Apple II和Apple II Plus電腦中，其實並未啓動第0槽。

2.6.3 IN＃指令

輸入指令IN＃係指定一設備，做爲輸入字元的來源。IN＃指令也可用來執行磁碟中的程式，只要指定插有您的磁碟控制卡的擴充槽即可。

IN＃指令的語法如下：

IN＃ n

其中n是1至7的數字。DOS連於指定槽(slot)的設備上讀取字元。數字記號＃是指令的一部分，必須要打入。

註：程式中的寫法是PRINT D$；" IN#n"

例如，如果您的Apple II電腦有一外接終端機連於第4槽，而且您希望您的程式從外接終端機讀入字元，可利用如下的指令：

```
10 D$=CHR$(4)
20 PRINT D$; "IN# 4"
```

在所有的Apple II電腦中，利用IN #0指令都可回復成從鍵盤讀取輸入（ input ）。

順便一提： IN #0是特殊情況，宣告 DOS 由鍵盤讀取輸入，其實並未啓動第0槽。

2.7 RENUMBER程式

重編號程式RENUMBER可將您的Applesoft BASIC 程式內所有指述或部分指述重新編號；將兩份程式的指述加以合併；或者將副程式館（ subroutine library ）的某一副程式（ subroutine ）插入您的現有程式中。重編號程式RENUMBER儲存於系統主磁碟（SYSTEM MASTER disk ）中。欲使此程式，必須將SYSTEM MASTER系統主磁碟片放入磁碟機，並打入以下指令：

```
RUN RENUMBER
```

重編號程式RENUMBER的標題畫面出現之後（如2−2 圖 ），按RETURN鍵，螢幕會出現 BASIC 的提示字元（ prompt character ）。這時您可以下任何RENUMBER指令，然而重編

註： 重編號程式RENUMBER是程式工具（ programming tool ）而非公用程式（ utility program ）。

號程式RENUMBER仍然留在記憶體中。您也可以繼續輸入或修改 BASIC 程式，您仍然可以執行或保存程式，就好像重編號程式RENUMBER不在身旁似的。

```
                APPLESOFT RENUMBER
        COPYRIGHT APPLE COMPUTER, INC. 1978

RENUMBER (DEFAULT VALUES)
&   [FIRST 10]  [ ,INC 10 ]  [ ,S 0 ]  [ ,E 63999 ]

MERGE
&H PUT PROGRAM ON HOLD
&M MERGE TO PROGRAM ON HOLD

PRESS 'RETURN' TO CONTINUE ...
```

圖 2 － 2 RENUMBER的標題畫面

2.7.1 RENUMBER指令

指令END、FIRST 、INC 和START用來指定重編號的範圍。其中 n 值爲 0 到 63999 的十進位數目。

START n　　(或 S n)其中 n 是記憶體中程式某一列的列號，此指令指明重編號的起始列。如果略去 S 指令，重編號程式RENUMBER將其值設定爲 0 ，亦即從程式的第一列開始。

END n　　(或 E n)其中 n 是記憶體中程式某一列之列號。此指令用來指明重編號的終止列。如果略去 E 指令，則重編號程式RENUMBER假設其值爲 63999 。

FIRST n　　(或 F n)其中 n 是十進位整數，此指令用來指定起始列(即 S 指令指定的那一列)的新列號(line number)。如果省略 F 指令，則重編號程式RENUM-

BER假設其值爲 10，亦即重新編號的程式之第一列編爲列 10 。

INCn　（或 In）其中 n 是增加量（ increment），此指令用來指定相隔兩列之間列號的差數。如果省略 I 指令，重編號程式RENUMBER設其值爲 10，所產生的各列順序爲 110 、 120 、 130 …。

HOLD　（或 H ）將您的程式放入駐留區（ hold buffer），並顯示 PROGRAM ON HOLD, USE '&M' TO RECOVER. 的字樣，表示程式已進入駐留區，如果您希望程式重回記憶體，則使用MERGE指令。

MERGE　（或M ）將記憶體中的程式和駐留區中的程式予以合併。M指令可將副程式館（ subroutine library）中的副程式（ subroutine ）插入您的程式中。

MERGE指令將各列以由小而大的順序加以排列。如果有指述列號相同時，MERGE 指令將兩者都放入最後的程式，將記憶體中程式的指述置於前頭，將駐留區程式的指述置於後頭。

如果記憶體並無程式，MERGE 指令將駐留區中程式回復到記憶體。如果您將程式放入駐留區之後，改變主意，您可以在尚未將其他程式載入記憶體之前利用M指令由駐留區將其還回記憶體。

如果駐留區中沒有任何程式而您使用M指令，重編號程式RENUMBER 會顯示 NO PROGRAM IN MEMORY 的字樣。

2.7.1.1 重編號指令的語法

重編號指令列的第一個字元是&（ ampersand ）。最短的指令列是：& (RETURN) 。當您使用此指令時，重編號程式RENUM-

BER即爲記憶體中的程式重新編號，由第一列指述到最後一列指述，第一列指述定爲列10，然後每個指述號碼（statement number）逐列加10。

當一行指令中含有一個以上的指令時，則各指令之間須以逗號（，）隔開，例如：& S 50,E 100,F 500告訴重編號程式RENUMBER處理由列50至列100的指述，並且將第一列的列號定爲500。

指令可簡寫，以第一個字元代表，指令得以任何順序出現。

2.7.1.2　內部"列號叫用"

重編號程式RENUMBER除了將您的程式指述（statement）重新編號之外，也會調整以下的指述之**列號叫用 (line-number reference)**：

GOTO	ON...GOTO
GOSUB	ON...GOSUB
DEL	THEN（如 IF X = 0 THEN 250）
RUN	

對於註解（remark）中的"列號叫用"（line-number reference），重編號程式RENUMBER不會加以修正，因此，如果您希望程式的註解（REM指述）正確反應您的程式內容，則必須自行將REMARK指述中的"列號叫用"加以修正。

註："列號"與"列號叫用"的意義及其差異，可舉下例說明之：10 IF A＞0 THEN 30 ELSE GOSUB 500，其中"10"即是該指述的"列號"而"30"及"500"則是指數中的"列號叫用"。假若您想重編號即是重新調整各列列號，接著各"列號叫用"也隨之一一修正成原值對應的新值。）

2.7.1.3 駐留區

當您使用HOLD指令時，重編號程式RENUMBER會將目前的程式放入 Apple II 記憶體中爲RENUMBER保留的區域保存起來。這塊區域稱做**駐留區**（hold buffer）。無論是 BASIC 或DOS都不會使用這塊區域。因此，如果您需要充分的記憶體時，儘量在需要合併程式（ merging program ）時，才使用這塊駐留區。

警告：重編號程式RENUMBER處理16K程式必須花一 分多鐘。
因此，當處理您的程式時，不要按[RESET]！要等到BASIC的提示符號] 出現後，才繼續下達下一個指令。
當重編號程式RENUMBER在執行時，按下 [RESET] 鍵或下達MERGE指令都可能毀掉您的程式。
當重編號程式在記憶體中時，不要使用MAXFILES指令，否則 DOS 會蓋到RENUMBER將其毀壞。
重編號指令（ RENUMBER command ）也可以放入指令檔（ command file)，但是執行重編號指令之後(用EXEC指令），重編號程式返回 BASIC ，而非指令檔（ command file ）。
對於指令CONVERT、HOLD、MERGE、重編號程式RENUMBER將其立即執行。因此，如果有任何指令接於指令CONVERT、HOLD或MERGE同一列之後，則RENUMBER不加以處理。

2.7.1.4 範例

第一個例子將整份程式重編號，第二個例子只對程式某部分予以重編，第三個例子則合併兩個程式。

註：有關指令檔（ command file ）詳見第 5 章。

1. 將您的整個程式予以重編，以列號 10 起頭，然後每一指述號碼（ statement number ）增加 10 。首先將 **SYSTEM MASTER** 系統主磁碟放入第一部磁碟機，並打入

```
RUN RENUMBER
```

然後，利用鍵盤輸入您的程式或利用載入指令**LOAD**將您的程式從磁碟搬入記憶體。畫面出現 **BASIC** 的提示字元] 之後，按下 & 然後是 **RETURN** 。利用 **LIST** 指令即可查看重編號後的結果。

2. 以下的樣本程式用來說明對程式的某一部分重編號。首先確定重編號程式RENUMBER在記憶體中，然後打入左邊的程式。

原程式

```
1  INPUT X
2  IF X<1 THEN 1
3  ON X GOSUB 39,87
27  END
39  PRINT A
45  RETURN
87  PRINT A * A
99  RETURN
```

重編號後的程式

```
1  INPUT X
2  IF X<1 THEN 1
3  ON X GOSUB 39,30
27  END
30  PRINT A * A
39  PRINT A
40  RETURN
45  RETURN
```

然後請打入以下的指令

```
& START 87, END 99, FIRST 30
```

只重編程式中的兩個指述。利用列印指令 **LIST** 看看重編後的程式。列 87 變成 30 ；列 99 變成 40；注意，RENUMBER已將列 3 中“列號叫用”（ line-number reference ）加以修正。

意外：這項技巧正好可用來將程式的某一段，從某一部分移到另一部分。

3. 假設您要合併兩份程式，如果重編號程式 RENUMBER 已在記憶體中，請載入（load）您的第一份程式，您可以加入新的指述，重新編號（renumber）執行程式以便測試，編輯此程式。當您滿意之後，打入以下的指令將其存入駐留區（hold buffer）：

```
& HOLD
```

現在載入第二份程式，同第一份一樣，您可以發展並測試此程式。將您第二份程式重編號使其不會與第一份程式相衝突。

最後，合併程式請打入以下指令：

```
& MERGE
```

如果您願意的話，還可以執行此新完成的程式，加以測試然後才存入磁碟。

2.7.2 RENUMBER 的錯誤訊息

```
> 63999
```

您打入的值超出範圍

```
DUPLICATE LINE NUMBERS
```

註：如果重編號程式 RENUMBER 執行時發現無可挽救的錯誤，會在更改您的程式之前停止執行，同時顯示一列錯誤訊息（error message），指示您的問題所在。

如果重編號程式RENUMBER繼續處理，則會有一新列號（ line number ）和已有的列號相同。因此，您必須用比較小的增量（ increment，**INCn**）或用不同的新起始列號（ **Fn** ）。

```
HOLD FILE IN USE
```

您使用第二個HOLD指令，而沒有先下MERGE指令。

```
LIMITED MEMORY, MAY DESTROY PROGRAM. CONTINUE
(Y/N)?
```

您系統太小或者您的程式太大，如果重編失敗，可能會毀掉您的程式。除Y鍵之外，打入其他的任何鍵，即可取消重編的工作並回到 **BASIC** 。

```
LINE INCREMENT = 0
```

此值不合法，必須指定大於 0 的值。

```
LINE INCREMENT TOO LARGE
```

您所指定的增量（ increment）使得某一列號超過 63999 。

```
LINE TOO LONG
```

重編號的結果使得某一列超過 239 個字元。請儘量減少各列的指述數量。

```
NO LINES IN RANGE
```

重編號程式RENUMBER找不到任何一列在您的指定範圍之內，沒有任何一列重新編號。

```
NO PROGRAM IN MEMORY
```

沒有程式在記憶體中，無法執行重新編號的工作。您可以利用&MERGE指令，將駐留區中的程式搬回記憶體中。

```
OUT OF MEMORY
```

處理工作需要有更多的記憶空間。重編號程式RENUMBER本身大約佔 2 K位元組（ bytes ）。

SYNTAX ERROR

重編號程式無法辨認指令的第一個字元，或者您輸入的值不合法。

2.8 摘 要

2.8.1 指令

FP [, S] [,D] [, V]

轉入Applesoft **BASIC** 語言系統，記憶體中的指標（ pointer ）重新設定，所以您無法找到原有的程式和變數。

INT

轉入 Integer **BASIC** 語言系統，記憶體中的指標（ pointer ）重新設定，所以您將無法再找到先前的程式和變數。

MON [C] [, I] [, O]

指示 **DOS** 開始顯示電腦與磁碟之間的任何動作或資訊。

NOMON [C] [, I] [, O]

由 **MON** 指令所開啓的顯示訊息，利用此指令關閉。

PR# n

指定連接於第 n 槽的設備為輸出字元的目的地。

IN # n

指定連接於第 n 槽的設備為輸入字元的來源處。

2.8.2 程式

RENUMBER

將 **BASIC** 程式的全部或一部分予以重新編號，或將兩程式的指述加以合併（merge）。

3 循序存取文字檔的使用

3.1 比較循序與隨意存取文字檔 55
3.1.1 選擇循序或隨意存取文字檔 57
3.1.2 目前位址指標 58
3.2 循序存取文字檔 59
3.2.1 資料欄 59
3.2.1.1 將字元存入資料欄 60
3.2.1.2 循序存取文字檔的簡短例子 61
3.2.2 輸入或讀取文字 63
3.2.2.1 利用PRINT寫入資料至檔案內 63
3.2.2.2 從檔案讀入字元 64
3.2.2.3 "各欄含一片段"的範例 66
3.2.2.4 "各欄含數片段"的範例 68
3.2.2.5 讀取含有逗號的資料欄 70
3.2.3 程式：輸入和讀取文字 72
3.2.3.1 輸入文字的程式 73
3.2.3.2 擷取文字的程式 75
3.3 循序存取檔的指令 77
3.3.1 OPEN指令—延遲執行模式 77
3.3.2 CLOSE 指令—立即或延遲執行模式 79
3.3.3 WRITE 指令—延遲執行模式 80
3.3.4 READ指令—延遲執行模式 81
3.3.5 POSITION 指令—延遲執行模式 82
3.3.6 APPEND指令—延遲執行模式 83
3.4 指令摘要 84

第3章

循序存取文字檔的使用

本章討論文字檔，並比較兩種不同的檔案類型：循序存取(sequential-access)和隨意存取(random-access)，然後描述如何寫程式和使用"循序存取文字檔"(sequential-access text file)的 DOS 指令。

文字檔(text file)是存於磁碟中一系列的資料欄(field)。**一資料欄(field)**係指一串由 1 至 32767 個字元組成，以"歸位字元"結束的資料。

檔案目錄(catalog)中檔案型態(file-type)該直行中用T字母標定此兩種檔案爲"文字檔"(T即是Text 的簡碼)。

3.1 比較循序與隨意存取文字檔

循序存取文字檔(sequential-access text file)係一系列**資料欄(field)**，各資料欄不定長。各個連續資料欄緊跟著上一個資料欄結束點的**歸位字元(carriage return)**之後繼續存放資料。每當 DOS 寫入或讀出循序存取文字檔時，DOS 係從檔案的第一個資料欄開始，然後一個接一個資料欄，依先後順序存取各個資料欄。

註：適用隨意存取文字檔的 DOS 指令，請參考第 4 章。

隨意存取文字檔（random-access text file）係一系列的**資料録 (record)**，各資料錄具有相同的長度，須預先設好各錄的字元數。因爲資料錄（record）可含有歸位字元，此字元用來表示資料欄（field）的結束，因此一資料錄（record）可包含一至數個資料欄（field），但是隨意存取檔案的所有資料錄均具有相同的長度。每當您開啓（open）隨意存取檔（random access）時，您必須指明資料錄長度（record length）。利用此方式，即可指示 DOS 存取檔案的任何欄位。

文字檔（text file）儲存 ASCII 碼（即文字）組成的字串，檔案中字串的放置係依據此文字以循序存取（sequential-access）指令或隨意存取（random-access）指令寫入而定。圖 3-1 舉例說明如何將文字放入循序存取檔；圖 3-2 則是隨意存取檔的例子。假設其資料錄長度是 5，每錄含一資料欄。

↵字元表示RETURN 字元，通常在大部份PRINT 指述會自動送出RETURN 字元（即“歸位字元”）。

字元：	7	↵	A	T	↵	O	N	E	↵	B	L	O	W	↵							
*ASCII:	B7	8D	C1	D4	8D	CF	CE	C5	8D	C2	CC	CF	D7	8D	00	00	00	00	00	00	00
檔案位元組號碼：	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
欄號：	0		1			2				3											

圖 3-1 輸出到循序存取文字檔

ASCII 值以 16 進位數字表示，注意其資料寫入磁碟時最高位元（bit）設定爲 1（譯註：在標準的 ASCII 編碼系統，通常是僅用 7 個位元來代表一個字元，最高位元則爲 0，例如字母 'A' 其對應的 ASCII 碼爲 41（十六進制），在Apple 文字檔內因爲規定最高位元設定爲 1，故 'A' 的ASCII 碼爲C1。）。

字元：	7	↵				A	T	↵			O	N	E	↵		B	L	O	W	↵	
ASCII：	B7	8D	00	00	00	C1	D4	8D	00	00	CF	CE	C5	8D	00	C2	CC	CF	D7	8D	00
檔案位元組號碼：	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
資料錄位元組號：	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0
欄號：	0					0					0					0					
錄號：	0					1					2					3					

圖 3-2　輸出到隨意存取檔案

ASCII 值以 16 進位數字表示，注意其資料寫入磁碟時最高位元（bit）設定為 1。

當您要將資料放入檔案之前，您的程式必須開啓（open）此檔。當您用完此檔案之後，您的程式必須將其關閉（close）。為了處理起始與結束的動作，可利用 DOS 指令OPEN與CLOSE。

當您的文字檔開啓後，您的程式即可輸出新的一列資料到檔案或者從檔案一列一列讀出資料。欲加入資訊於文字檔，可利用DOS指令WRITE和 BASIC 的PRINT 指述；欲擷取資料則利用DOS的指令READ和 BASIC 指述 INPUT 和GET。

在指令中可選用的引數（argument），視您使用那一種型式的檔案而定，主要的不同是隨意存取指令須指定資料錄長度。

3.1.1　選擇循序或隨意存取文字檔

循序存取檔的資料欄可以是不定長，您只能循序使用各資料欄，，從檔案的最開頭起，往結尾的方向一一取用。某些應用係將檔案的整個內容讀進記憶體，加以修改後再將修改後的內容寫回檔案，用此種方式使用檔案，則循序存取是較佳的檔案型式。很多文書處理程式（word processor）將其文字存入循序存取文字檔。

在隨意存取文字檔中，所有資料錄（record）必須具有相同長度。您可以不依先後次序，任意讀取某一資料錄或修改任何資料錄，而不至於影響其他資料錄的儲存位址。如欲儲存很多片段資訊，其最大長度已確定，同時其內容經常修改，則較佳的型式是隨意存取檔。例如：您可將股票價格或實驗數據存入隨意存取檔。

在選擇使用何種型式的文字檔時，您必須考慮以下的因素：

- **磁碟空間：** 當您首度將一資料錄寫入隨意存取文字檔時，整個資料錄即放入檔案中。如果您指定的資料錄長度是200個字元（characters），而每個資料錄僅寫入一字元，則您每個資料錄即浪費199個字元的磁碟空間（198加上carriage return）。在眞正使用的狀況，很少有資料錄是完全塡滿的。因此，隨意存取文字檔使用磁碟空間比循序存取文字檔要浪費些。
- **資料數量：** 當您開始執行程式時，將所有資訊讀入記憶體，則從循序存取文字檔中一欄接一欄讀取資料，比隨意存取文字檔快。
- **資料的使用方式：** 如果資料不移入記憶體，而且您不以某種特定次序使用它，則使用隨意存取文字檔。

3.1.2 目前位址指標

每個開啓（open）的檔案都有一指標來追踪檔案中的**目前位址**（**current position**），本書我們稱之爲**目前位址指標**（**current**

註：任何文字檔的結尾都自動標定爲16進位$00值。

註：有關循序存取指令的語法和使用，請參考第3.3節。

註：有關隨意存取指令的語法和使用，請參考第4.3節。

position pointer)。當您從檔案讀入資料錄，則最後一個字元讀入之後，目前位址指標指向 carriage return 之後的字元。同樣地，當您將資料錄寫入檔案之後，則指標指向最後一個 carriage return 緊接的位址。當您開啓檔案時，目前位址（current position）是檔案中第一個字元的位址。

3.2 循序存取文字檔

循序存取文字檔 (sequential-access text files) 就好比一個卷軸，含有無限列的文字。使用循序存取檔或卷軸，您必須一列一列找才能得到某一特定文字列，並沒有分頁之類的大段落讓您快速搜尋。

3.2.1 資料欄

循序存取文字檔的基本單位是**資料欄 (field)**。一資料欄可比擬成螢幕上的一列文字，兩者都是一系列的字元，以 carriage - return 字元結束。欲瞭解如何在文字檔中建立資料欄，首先去觀察 BASIC 的 PRINT 指述，如何將一資料欄送至螢幕上。

當 PRINT 指述輸出一列文字到螢幕上時，只要不是用分號（semicolon;）結束此指述，則 BASIC 自動在此列結尾放入 carriage return 字元，您會看到游標（cursor）移到下一列。當您利用 PRINT 指述輸出到檔案時，只要此指述不以分號結束，則 BASIC 將 PRINT 的字串內容放入檔案中，並在字串結尾寫入 carriage return 字元。在檔案中，“字串”叫做“資料欄”。下一個 PRINT 指述則產生檔案的下一個資料欄。

以下的 BASIC 指述可用以將一列文字寫入螢幕或者將一資料

欄寫入檔案。

```
50 PRINT "THIS CREATES A FIELD"
```

分號用來告訴 BASIC ，不要在此列（或此欄）的結尾加入 carriage return 字元。如果此行或此欄還有更多的資料待寫，則在 PRINT 指述的字串之後加上分號，例如：

```
100 PRINT "THIS MAKES PART OF A FIELD";
```

由於列 100以分號結束，因此下一個 PRINT 指述會將其字元加入螢幕上同一列之後（或者是檔案的同一資料欄）：

```
110 PRINT " AND OTHER PARTS WILL FOLLOW."
```

則螢幕上，列 100 和列 110執行結果如下：

```
THIS MAKES PART OF A FIELD AND OTHER PARTS WILL
  FOLLOW.
```

3.2.1.1 將字元存入資料欄

看看以下的PRINT 指述：在檔案中產生數個資料欄，同時顯示各資料欄在循序存取資料檔（sequential - access text file）中的關係。

```
40 PRINT "GREEN"
50 PRINT "YELLOW"
60 PRINT "ORANGE"
70 PRINT "RED"
80 PRINT "VIOLET"
90 PRINT "BLUE"
```

此循序存取文字檔含有 36 個字元，存於不同長度的 6 個資料欄。如果輸出到畫面上，則列 40 至列 90 執行結果如下：

```
GREEN
YELLOW
ORANGE
RED
VIOLET
BLUE
```

以下是列 40 至 90 存入檔案的結果：

字元次序：GREEN) YELLOW) ORANGE) RED) VIOLET) BLUE)
欄　　號：0　　1　　2　　3　　4　　5

順便一提：注意，循序存取文字檔的第一欄編號為 0 。

3.2.1.2 循序存取文字檔的簡短例子

樣本程式 **LISTSELF** 建立一循序存取文字檔，叫做 **LISTFILE**，並將各欄資料存入檔案。

```
10 REM PROGRAM LISTSELF
20 D$ = CHR$ (4)              （D$是CONTROL-D）
30 PRINT D$;"OPEN LISTFILE"
                              （開啓LISTFILE；如果
                              不存在，開啓此檔案。）
40 PRINT D$;"WRITE LISTFILE"
                              （準備寫入LISTFILE）
50 LIST                       （將程式列印輸入檔案；列
                               40指定的檔案。）
60 PRINT D$;"CLOSE"           （關閉所有開啓的檔案）
```

這個簡短的程式指示 **DOS** 開啓（open）檔案 **LISTFILE**（列 30 的指述），利用 **WRITE** 指述指示將資料寫入檔案 **LISTFILE**（列 40 的指述），執行 **BASIC** 的 **LIST** 指述，然後關閉（close）**LISTFILE** 檔。

註：) 符號代表 carriage return 字元（歸位字元）。

注意，LIST 不是 DOS 指令，不可將 CONTROL - D 置於前頭。事實上，如果您輸出 D$ ，則將會取消上一個 DOS 指令，即列 40 的WRITE 的指令。

WRITE 指令將所有輸出指向檔案 LISTFILE ，因此，LIST 指述將程式一列接一列寫入 LISTFILE 檔案，而不是送到螢幕上。

打入此程式，然後將已啓始（initialized）且可寫的磁碟放入第一部磁碟機，並打入

```
RUN
```

此指令會執行 LISTSELF ，於是將名爲 LISTFILE 的文字檔存入磁碟中。在 LISTSELF 程式執行完畢之後，提示字元] 會再度重現螢幕，再看看磁碟目錄，您會發現 LISTFILE 在裏面，佔用兩個磁區（sector），標定爲T表示文字檔。欲檢查新的文字檔的內容，您可以利用EXEC指令。EXEC指令會處理文字檔，並讀出指令來執行，而不從鍵盤讀取指令。

當您打入數列的 BASIC 程式，則此程式存入記憶體。同理，如果您利用EXEC指令從循序存取文字檔讀入程式，則也同BASIC 程式一樣存入記憶體。您可以利用 LIST 指述來查看記憶體內的 BASIC 程式。

順便一提： 立即執行模式（Immediate - execution）的 BASIC 指令係執行記憶體中的程式。此程式可能是您打入的程式，或是利用LOAD指令從磁碟載入，或是將EXEC檔案帶入記憶體。

首先用 NEW 指述清除記憶體，然後利用 LIST 指述證明沒有任何程式在記憶體中，然後打入指令

註： 請參考第 5 章的EXEC指令。

```
EXEC LISTFILE
```

螢幕上 **BASIC** 程式的每一列都會出現提示字元]。當磁碟停止轉動時，**BASIC** 會輸出提示字元，不接任何程式的指述。這時您可以打入

```
LIST
```

LIST 指述會列出重回記憶體的程式。

3.2.2　輸入或讀取文字

文字寫入檔案的方式，會決定以後如何從檔案讀取資料。以下是一系列相關的檔案使用舉例。

以下兩個研討中，以逗號分開的資料，稱做“**片段**”(**part**)。表 3-1 的指述 5（statement 5）說明如何建立片段；表 3-2 的指述 2 則說明如何讀取這些片段。

3.2.2.1 利用PRINT寫入資料至檔案內

PRINT 指述結束的方式，會影響到字元存入文字檔的方式。表 3-1，變數 A$ 存有值“DOG”；變數 B$ 存值是“CAT”；⤸ 字元則表示歸位字元 (carriage return)。

註：**NEW** 指述用來清除記憶體和變數。

表 3-1 輸出資料到文字檔

PRINT 指述	輸出字元	說 明
1. PRINT "TEXT"	TEXT↲	結束此欄並加入歸位字元。
2. PRINT "TEXT";	TEXT	不加入歸位字元。
3. PRINT A$;B$;	DOGCAT	將第二欄資料併入第一欄，不加入歸位字元。
4. PRINT A$,B$	DOGCAT↲	與輸送到螢幕不同，不在各欄之間加入空白，但結尾附上歸位字元。
5. PRINT A$;",";B$	DOG,CAT↲	將逗號（，)和第二欄資料一併加入第一欄，且附上歸位字元。

3.2.2.2 從檔案讀入字元

您如何利用 PRINT 指述將字元放入檔案，會決定讀出資料的方式。某種型式的資料比較適合用 INPUT 來讀取其值，其他型式的資料則用 GET 比較合適。

INPUT 指述的變數個數須與資料欄內“片段”的個數相對應。假若 INPUT 指述僅含有一個變數，則讀出目前所在資料欄結尾的歸位字元（carriage return）之前的所有字元，但是只將冒號（：）或逗號（，）之前的各字元放入此變數。因為 INPUT 可將資料讀入數個變數中，只要各“片段”資料之間以逗號分開，即可增加 INPUT 的變數個數來讀取該欄第一個逗號之後的字元（每一變數相對應於一個“片段”）。

利用 GET 指述可讀取在 INPUT 指述所無法讀取的資訊。GET 指述可讀取任何的字元，包括逗號、冒號以及控制字元（control character）。您可以利用這種方式一次一個字元讀出任何

不定長的各片段之內容。

注意： INPUT 指述僅能讀到 239個字元，超出的部份會去掉。如果資訊超出 255字元，則 INPUT 取消整列並重新開始。

表 3-2說明利用 INPUT 和GET讀取字元的一些例子。

表 3-2　從文字檔讀取資料

指　　述	效　　果
1. INPUT A$	讀取一資料欄的第一個片段。如果此資料欄有一個以上的片段，BASIC 會捨棄其餘的片段，並顯示如下的訊息：EXTRA IGNORED，表示該欄尚有其他"片段"資料，但無變數與之對應，無法讀入它們。
2. INPUT A$,B$	讀取資料欄的兩個片段。如果此欄有二個以上片段，BASIC 捨棄其他的片段，並顯示如下的訊息：EXTRA IGNORED。如果此資料欄少於兩個片段，則 BASIC 讀取下一個資料欄的片段。
3. GET C$	從檔案讀出下一個字元（字母、逗號或冒號）。

以下三個例子，分別是"各欄含一個片段"，"各欄含數片段"，以及"資料欄含有逗號"，分別說明如何輸入與讀出文字的程式技巧。

註： 請參考第 3.2.2.5節中，有關 GET 的警告。

3.2.2.3 “各欄含一片段”的範例

本程式說明如何將四個資料欄（field）寫入循序存取文字檔，其中各欄僅含一片段（parts）。

```
10 REM MAKE.FRUIT
20 D$ = CHR$ (4)            （D$是CONTROL-D）
25 PRINT D$;"MON C,I,O"     （觀察操作過程）
30 PRINT D$;"OPEN FOUR.FRUITS"
                            （必要時，建立FOUR.
                            FRUITS 檔案；並予以開啓。）
40 PRINT "THIS GOES TO THE SCREEN"
                            （字元進入螢幕，等到WRITE
                            執行之後，才寫入檔案。）
50 PRINT D$;"WRITE FOUR.FRUITS"
                            （指定輸出檔案）
60 PRINT "APPLE"            （寫入第1欄至檔案）
70 PRINT "BANANA"           （寫入第2欄至檔案）
80 PRINT "CANTALOUPE"       （寫入第3欄至檔案）
90 PRINT "DATE"             （寫入第4欄至檔案）
92 PRINT D$;"CLOSE FOUR.FRUITS"
                            （關閉FOUR.FRUITS）
94 PRINT D$;"NOMON C,I,O"
                            （停止監視動作）
96 END
```

注意，即使檔案**FOUR. FRUITS**已開啓，第一列輸出仍然是到螢幕（列40）。但是，列50的**WRITE**指述之後，所有的**PRINT**指述都將字元送到檔案上。

以下是字元存入檔案**FOUR. FRUITS**的結果：

```
字元次序： APPLE) BANANA) CANTALOUPE) DATE)
欄    號： 0      1       2           3
```

以下的程式從**FOUR·FRUITS**中讀入此四個資料欄，僅使用一個**INPUT**指述，並將其顯示出來，表示操作正常。

註：) = carriage return （歸位字元）

```
10 REM GET.FRUIT WITH ONE INPUT STATEMENT
20 D$ = CHR$ (4)              ( D$是CONTROL-D )
25 PRINT D$;"MON C,I,O"  ( 監視操作 )
30 PRINT D$;"OPEN FOUR.FRUITS"
                              ( 準備使用FOUR.FRUITS )
40 PRINT D$;"READ FOUR.FRUITS"
                              ( 指定檔案做爲輸入 )
50 INPUT A$(0),A$(1),A$(2), A$(3)
                              ( 如果某資料欄沒有多的片段
                              ,DOS由下一個資料欄讀入第
                              一片段。 )
80 PRINT A$(0)" "A$(1)" "A$(2)" "A$(3)
                              ( 水果名稱輸出於螢幕上 )
90 PRINT D$;"CLOSE FOUR.FRUITS"
                              ( 結束關閉FOUR.FRUITS )

94 PRINT D$;"NOMON C,I,O"( 停止監視 )

96 END
```

您會看到螢幕上顯示

```
APPLE BANANA CANTALOUPE DATE
```

以下的程式則將循序存取文字檔的四個資料欄，讀入**陣列 (array)** 的連續四個元素。列 1 5 的 **DIM** 指述其目的是宣告陣列 (array)所含有**元素 (element)** 的數量，在記憶體內保留該陣列需用的空間。

```
10 REM GET.FRUIT.INTO.CUPS USING AN ARRAY
15 DIM A$(3)
20 D$ = CHR$ (4)              ( D$是CONTROL-D )
25 PRINT D$;"MON C,I,O"  ( 監視操作 )
30 PRINT D$;"OPEN FOUR.FRUITS"
                              ( 準備使用FOUR.FRUITS )
40 PRINT D$;"READ FOUR.FRUITS"
                              ( 準備讀檔案 )
```

註: 陣列 (array)係由元素 (elements, 數字或文字) 所組成的表格 (table)。

```
50 FOR I = 0 TO 3          (由第 0 欄至第 3 欄依次讀
60 : INPUT A$(I)           取檔案中的資料欄輸出到螢
70 : PRINT A$(I)           幕上)
80 NEXT I                  (然後處理下一個資料欄)
90 PRINT D$;"CLOSE FOUR.FRUITS"
                           (結束，關閉FOUR.FRUIT)
```

此程式對於每一資料欄，都以一個 PRINT 指述來讀取內含資訊。出現在列 6 0 和列 7 0 的冒號並非必要的，只是使程式更容易閱讀，表示進入一層迴路 (loop)。

3.2.2.4 "各欄含數片段"的範例

以下的程式將兩個資料欄各放入三個片段。請將此程式打入記憶體，並加以執行。在列 5 0 和列 6 0 中，逗號之間的資料也會寫入檔案 INVENTORY，因爲此字元在雙引號 (quote")之內。

```
10 REM BINS
20 D$ = CHR$ (4)              (D$是CONTROL-D)
25 PRINT D$;"MON C,I,O"       (監視操作)
30 PRINT D$;"OPEN INVENTORY"
                              (必要的話，建立檔案INV-
                              ENTORY;準備使用此檔)
40 PRINT D$;"WRITE INVENTORY"
                              (準備寫檔案)
50 PRINT "BOLTS,SCREWS,NAILS"
                              (寫入第 0 欄的 3 個片段)
60 PRINT "WASHERS,GROMMETS,RINGS"
                              (寫入第 1 欄的 3 個片段)
70 PRINT D$;"CLOSE INVENTORY"
                              (關閉 INVENTORY)
94 PRINT D$;"NOMON C,I,O"
                              (停止監視)
96 END
```

以下是字元存於INVENTORY 的方式：

字元次序：BOLTS,SCREWS,NAILS)WASHERS,GROMMETS,RINGS)
欄　　號：0　　　　　　　　　　1

下兩個程式中的 **INPUT** 指述以不同的方式來讀取這些逗號之前的“片段”內容。程式讀取資料時，以逗號做爲各片段的結束。

此程式將各片段讀出，並存入對應的各個不同變數。

```
10 REM DRAWERS
20 D$ = CHR$ (4)              ( D$是CONTROL-D )
25 PRINT D$;"MON C,I,O"  ( 監視操作 )
30 PRINT D$;"OPEN INVENTORY"
                              ( 準備使用INVENTORY )
40 PRINT D$;"READ INVENTORY"
                              ( 準備讀檔案 )
50 INPUT A1$,A2$,A3$,A4$,A5$,A6$
                              ( 讀取第0欄的3片段與第1
                              欄的3片段 )
60 PRINT D$;"CLOSE INVENTORY"
                              ( 關閉INVENTORY )
70 PRINT A1$,A2$,A3$,A4$,A5$,A6$
                              ( 顯示所有6個片段 )
80 PRINT D$;"NOMON C,I,O"
                              ( 停止監視 )
90 END
```

注意，列50只是從檔案讀入連續的片段，當資料欄中所有的片段(parts)都讀完之後，會自動從下一個資料欄讀入其他的片段。

以下的程式只讀入各資料欄的第一片段。

```
5 REM METAL.PARTS
10 D$ = CHR$ (4)              ( D$是CONTROL-D )
20 PRINT D$;"MON C,I,O"  ( 監視操作 )
30 PRINT D$;"OPEN INVENTORY"
                              ( 準備使用INVENTORY )
40 PRINT D$;"READ INVENTORY"
                              ( 準備讀檔案 )
50 INPUT A1$                  ( 讀第零欄的第1片段 )
60 INPUT A2$                  ( 讀第1欄的第1片段 )
70 PRINT D$;"CLOSE INVENTORY"
                              ( 關閉INVENTORY )
80 PRINT A1$,A2$              ( 顯示此兩個片段 )
90 PRINT D$;"NOMON C,I,O"
                              ( 停止監視 )
100 END
```

本例中，每個 INPUT 指述都讀入整個資料欄，而不管此欄含有有幾個片段。列 50 將 "BOLTS" 值指定給 A1$，捨棄第 0 欄的其餘部份，並顯示訊息：EXTRA IGNORED。同理，列 60 將 "WASHERS" 指定給 A2$，不管第 1 欄的其他部份，並顯示 EXTRA IGNORED 。最後，列 80 顯示 BOLTS 和 WASHERS 兩片段。

3.2.2.5 讀取含有逗號的資料欄

欲讀入逗號（，）、冒號（：）或控制字元（control character），或欲讀入並檢查特殊的字元，則不能用 INPUT 指述，須改用 Applesoft 的 GET 指述。INPUT 係讀入以逗號隔開的字串（string），但是 GET 則是從鍵盤或文字檔中一次讀取一個字元。

警告: 利用 Applesoft 的 GET 指述由文字檔讀入一個字元後，DOS 會產生以下的問題：

如果 GET 之後第一個輸出項目是 DOS 指令，則此DOS指令可能無法執行，因爲前頭的 carriage - return 字元不見了。要解決此問題，必須在輸出 DOS 指令之前，輸出一個 carriage return 。

當NOMON C,I,O生效時，在 GET 之後輸出的第一個字不會出現於螢幕上。

當MON C, I, O生效時，在 GET 之後輸出的第一個字元會出現於螢幕上。

欲克服這些問題，可在送到螢幕顯示的第一個字元之前頭，先送出“無法顯示的字元”（nonprinting character 如 CONTROL - A ）。

注意，GET在讀取資料欄時，係一個字元接一個字元讀的，並存起來，直到歸位字元的出現表示資料欄結束。亦即是您利用 GET 來讀一長串的字串（string）或一串列合併的字

串(concatenated string),可能會造成記憶體不夠的問題。

您已做過一些例子,已輸入程式BINS並建好檔案INVENTORY。下一個例子係利用 GET 指述,從檔案INVENTORY擷取字元,GET指述從資料欄中讀取一個以上的片段。

INVENTORY每個資料欄各含三個片段。但是,通常我們可能並不知道一資料欄含有幾個片段。利用 GET 指述即可讀出未知數量的片段,因爲它藉助於資料欄中的逗號(,)將各片段分開。

以下的副程式(subroutine),辨別出逗號,讀入各片段,然後將其放入“字串陣列”(string array)A$的連續元素(element)上。任何時刻,使用中的元素是A$(I)。

副程式(subroutine)利用 GET 指述,讀入一個字元,存入變數C$。如果此字元不是逗號也不是歸位字元,則副程式將其存入A$(I)。如果副程式讀到逗號,表示此片段結束,則將變數I加1,因此I就指到陣列的下一個元素,然後繼續讀其他字元。此過程反覆執行,直到遇上歸位字元爲止,表示此欄結束。

```
10 R$ = CHR$ (13)           (R$是carriage return)
999 REM READ A FIELD
1000 I = 0                  (由陣列的第0元素起)
1010 I = I + 1              (使用陣列下一個元素)
1020 GET C$                 (讀入下一個字元)
1030 IF C$ = "," THEN GOTO 1010
                            (如果是逗號,使用下一個元素)、
1040 IF C$ = R$ THEN RETURN
                            (如果GET 讀入carriage re-
                            turn, 則表示沒有其他元素可讀,
                            否則將C$ 加入元素中。)
1050 A$(I) = A$(I) + C$

1060 GOTO 1020
```

要利用此副程式,必須有一程式來叫用此副程式,以便從檔案INVENTORY的第一個資料欄中擷取各片段。召用副程式,係利

用**GOSUB n**，其中 n 是副程式開頭第一個指述的列號（以下程式的 60）。當**RETURN**指述執行時（上個程式列1040），執行程序轉入GOSUB 指述（下個程式的列 70）。

```
20 REM USE SUBROUTINE
30 D$ = CHR$ (13) + CHR$ (4)
                              （D$是carriage return 加
                              上CONTROL-D ，以保證
                              CLOSE 會執行。）
40 PRINT D$;"OPEN INVENTORY"
                              （準備使用INVENTORY）
50 PRINT D$;"READ INVENTORY"
                              （準備讀出INVENTORY）
                              reading)
60 GOSUB 1000                 （讀入資料欄的所有片段）
70 PRINT D$;"CLOSE INVENTORY"
                              （關閉INVENTORY）
80 FOR J = 1 TO I             （現在將A$ 中的I個元素顯示
                              於螢幕上）
90 : PRINT A$(J)
100 NEXT J
110 END
```

您可以檢驗此程式，請輸入以上的主程式（Program）和副程式(subroutine)，打入

```
RUN
```

您會看到檔案**INVENTORY**的第一個資料欄內三個字語（word）顯示於螢幕上，每列一個字語。

3.2.3 程式：輸入和讀取文字

SAMPLE PROGRAMS磁碟中有兩個程式，可用來說明如何利用 **GET** 從檔案讀取不同數量的資料欄。程式**MAKE TEXT**從

註：**GET TEXT** 程式只讀出檔案的第一個資料欄。

鍵盤讀入文字，並將其存入檔案。程式**GET TEXT**則從檔案讀取文字，並將其顯示於螢幕上。

3.2.3.1 輸入文字的程式

以下的程式存於檔案**MAKE TEXT**中。此程式允許您最多輸入100列文字，並將其存入檔案中。其執行順序是，先要求輸入幾列文字，將其由鍵盤讀入，並放入陣列A$ 的連續元素上。當程式遇到空白列時，即停止讀入資料。**MAKE TEXT** 的第一部份大致如下：

```
5 REM    MAKE TEXT
10 DIM A$(100)                  ( 允許陣列存入100列文字 )
                                array)
20 D$=CHR$(4) : REM CONTROL-D
                                ( D$是CONTROL-D )
                                variable D$)
30 R$=CHR$(13) : REM RETURN
                                ( R$是RETURN )
40 TEXT : HOME                  ( 設成文字狀態；並清除畫面。 )
50 PRINT "      TEXT FILE CREATOR"
60 INVERSE : PRINT "     TO ADD A STRING:"
70 NORMAL : PRINT "     ENTER CHARACTERS, AND
     PRESS RETURN"
80 INVERSE : PRINT "TO END:"
90 NORMAL : PRINT "     PRESS RETURN ON AN
     EMPTY LINE"
100 PRINT : POKE 34,6           ( 跳過一列；指定由第6列爲畫面
                                顯示的上限，以上的部份供程式的
                                指令利用。 )
110 I = I + 1 : PRINT I;": ";
                                ( 要求輸入下一列文字 )
120 GOSUB 1000                  ( 跳至讀文字的副程式 )
130 IF A$(I) < > "" GOTO 110
                                ( 如果A$(1)不是空白列，跳
                                至列110，處理下一列。 )
```

註：TEXT指述將顯示畫面轉成 24 列文字狀態（text mode），而非畫圖狀態（graphics mode），並將游標置於最底列的開頭。

列 40 中，TEXT 指述將畫面轉入文字狀態（text mode），HOME指述則清除畫面上所有字元，並將游標移至畫面左上角。列 50 到列 110 將程式的提示指令顯示於螢幕上，某些字母正常顯示，某些則以反白字元（inverse letters）顯示。列 100 設定“文字窗”（text window），意即將前 6 橫列文字凍結於螢幕上，使提示指令留在畫面上。變數 I 在第一次使用時是 0，當程式第一次執行到列 110，將 I 設成 1，並將 1 輸出到螢幕上。

列 120 指示MAKE TEXT跳至副程式，以便將文字從鍵盤讀入。此副程式如下：

```
1000 GET C$: PRINT C$;      (讀入一個字元，並將其顯
                             示於螢幕上。)
1010 IF C$=R$ THEN RETURN
                            (如果此字元是RETURN
                            ，跳出此副程式；回到130)
1020 A$(I) = A$(I) + C$     (如果是其他字元，則將其存
                            入字串。)
1030 GOTO 1000              (回頭再讀以下的輸入字元)
```

MAKE TEXT讀入一個字元，並將其存入變數C$，同時檢查C$。如果 C$ 含有歸位字元，執行順序會跳回列 130。但是，如果 C$ 不含有歸位字元，則程式將 C$ 的內容加入陣列中。（列1020）程式將各個字串放入陣列 A$ 的第 I 個元素。

當文字存入陣列之後，MAKE TEXT 會詢問，這些文字將存入那個檔案，請輸入檔案名稱。

```
140 INPUT "SAVE TO WHAT FILE? ";N$
150 IF LEN (N$)=0 THEN 220
                            (如果沒有給予任何檔名，跳
                            到 220，放棄此資料。)
160 PRINT D$;"OPEN ";N$     (開啓檔名為 N$的檔案)
170 PRINT D$;"WRITE ";N$    (準備寫檔案)
```

註：有關設定文字窗的技巧，請參考Applesoft BASIC Programmer's Reference Manual。

程式將檔名存入變數 N$。下一列指述檢查是否檔名已確實輸入。如果檔名沒有輸入，則程式轉入列 220，然後放棄執行。

下個步驟，程式將陣列的內容，存入指定的檔案。空白列（即不打入任何文字，祇按 RETURN 鍵）指示文字的結束。然後，MAKE TEXT將整個陣列輸出到檔案（列 180至列 200）。

最後，MAKE TEXT利用列 220重設畫面指標（screen pointer），使其回到適當的文字狀態（text mode），並結束此程式。

```
180 FOR J = 1 TO I - 1 （對於每一列文字，將
190 : PRINT A$(J)        其寫入檔案，繼續下一
200 NEXT J               列。）
210 PRINT D$;"CLOSE";N$（完成後，關閉檔案。）
220 TEXT : END           （重設畫面爲正常大小）
```

執行此程式數次，建立不同長度的文字檔，實驗此程式的功能，並熟悉其工作的方式。例如，打入一列空白字元（space），將空白列輸入文字檔；因其含有“空白字元”，所以該列長度不是 0。

3.2.3.2 擷取文字的程式

建立循序存取文字檔是一項具有挑戰性的工作。您必須有一些方法，來擷取您放入檔案的資訊。

程式GET TEXT讀出循序存取文字檔。將程式從 SAMPLE、PROGRAMS磁碟讀入記憶體，看一看以下所描述的各列指述。

首先，GET TEXT 爲使用的變數設値；將陣列 A$ 的大小定爲 100個元素；設定 D$ 的値爲 CONTROL - D 及 R$ 的値爲 RETURN 。在執行工作之前，程式先清除畫面，將游標移到畫面左上角（列 50）。

註：如果檔名不以字母開頭，則 DOS 顯示訊息：SYNTAX ERROR.

```
5 REM GET TEXT
10 ONERR GOTO 150
20 DIM A$(100)
30 R$ = CHR$ (13)          ( R$是RETURN )
40 D$ = CHR$ (4)           ( D$是CONTROL-D )
50 TEXT : HOME
```

列 60 中，**GET TEXT** 詢問檔案名稱，以便由此讀取文字，並將此檔名讀入變數N$。列 70 告訴您，只要鍵入 CONTROL-S 即可停止程式執行，只要再按任何鍵即可恢復輸出。

```
60 INPUT "LIST WHAT TEXT FILE? ";N$
70 PRINT : PRINT "TYPE: CONTROL-S TO STOP
     LISTING"
80 PRINT "                  ANY KEY TO CONTINUE"
90 PRINT : PRINT : POKE 34,6
```

與上個程式一樣，**GET TEXT** 凍住畫面上端 6 橫列文字（列 90），因此即使您輸入的文字超出螢幕所能容納，上端的凍結畫面仍會保留在螢幕上。

讀入檔名之後，**GET TEXT** 隨即開啓指定的檔案，並由此檔案讀入連續的數列文字。

```
100 PRINT D$; "OPEN "; N$
                            ( 開啓檔名 N$ 的文字檔 )
110 PRINT D$; "READ "; N$
                            ( 準備讀檔案 )
120 FOR I = 1 TO 100        ( 對於檔案每一列，跳入副程
130 GOSUB 1000              式處理每一個字元。 )

135 : PRINT A$(I)           ( 將各個字元顯示於畫面上 )

140 NEXT I                  ( 然後做下一個字串 )
```

最後的工作是關閉檔案，重設畫面。

```
150 PRINT D$;"CLOSE ";N$ (關閉檔案)
160 TEXT : END
```

以下是列 130 所叫用的副程式，非常類似於MAKE TEXT的副程式。

```
1000 GET C$: PRINT C$;   (讀入一字元，並顯示於畫面上)
1010 IF C$=R$ THEN RETURN
                          (如果字元是RETURN跳出此副
                          程式，回到135。)
1020 A$(I) = A$(I) + C$  (如果是其他字元，將其存入字串
                          中。)
1030 GOTO 1000            (再讀下一個字元)
```

3.3 循序存取檔的指令

本節描述處理循序存取檔的 DOS 指令。注意，這些指令中只有一種係立即執行（immediate execution）和延遲執行（deferred execution）模式都適用。其他的指令只適用於延遲執行模式。

3.3.1 OPEN 指令——延遲執行模式

在讀出或寫入資料於循序存取檔之前，必須先開啓（open）此檔。

當程式開啓一個文字檔時，DOS會預留 595 位元組（bytes）的記憶體空間，供此檔案利用。這塊空間稱做**檔案保留區（file buffer）**。同時 DOS 將目前位址（current position）指向檔案的開頭。

OPEN指令允許 16 個檔案同時開啓，但是必須利用指令MAXFILES設定最多檔案數，因爲每個 DOS 檔案都需要各自的檔案

保留區。

OPEN指令的一般格式是:

OPEN fn [, Sn] [, Dn] [, Vn]

fn 指示欲開啓的檔案名稱，如果此檔不存在，DOS利用此名稱建立一新檔，檔案型態爲T（文字檔）。如果檔案已存在，既然已開啓，則不能再開啓。

[, Sn] 其中n是0至7的數值，指示所欲使用磁碟機之磁碟機控制卡（disk controller card）所在的擴充槽（slot）。如果省略不指明，則 DOS 使用預設的槽號（default slot number）。

[, Dn] 其中n是1或2，指示放置存有 fn 檔案磁碟片的磁碟機（drive）。如果省略不指明，則 DOS 使用預設磁碟機號碼（default drive number）。

[, Vn] 其中n是0至254的數值，指定所用磁碟片的卷號（volume number）。如果您指定0，即只打V，或省略此選擇項，則 DOS 不管卷號

上一節已有不少OPEN指令的例子，本節告訴您一些新的OPEN使用例子。

假設您的程式經常將舊的文字檔以相同檔名的新版本取代之。如果新檔比舊檔短，則新檔的尾巴會有部份舊檔的殘存資料，除非程式先將舊檔删除。如果您不希望多餘的文字留在檔案尾部，則在寫資料入新檔之前，先將舊檔删除。下個程式告訴您如果删除檔案，並爲新資料開啓此檔，無論此檔案存在與否，皆可適用。

註：請參考第5章的MAXFILES指令。

```
5 REM DEMONSTRATES ERROR FREE DELETION
10 D$ = CHR$ (4)              ( D$是CONTROL-D )
20 N$ = "GAMES"               ( N$ 是檔名 )
30 PRINT D$;"MON C,I,O"       ( 監視操作 )
40 PRINT D$;"OPEN ";N$        ( 開啓檔案；如果不存在，
                              則建檔。 )
50 PRINT D$;"DELETE ";N$      ( 檔案確定已存在，可將其
                              刪除。 )
60 PRINT D$;"OPEN ";N$        ( 開啓檔案 )
```

以下的指述合成完整的程式。

```
70 PRINT D$;"WRITE ";N$       ( 準備寫檔案 )
80 PRINT "TIC"
90 PRINT "TAC"
100 PRINT "TOE"
110 PRINT D$;"CLOSE ";N$      ( 關閉檔案 )
120 PRINT D$;"OPEN ";N$       ( 開啓檔案 )
130 PRINT D$;"READ ";N$       ( 準備讀檔案 )
140 INPUT A$,B$,C$
150 PRINT A$,B$,C$
160 PRINT D$;"CLOSE ";N$      ( 關閉檔案 )
170 PRINT D$;"NOMON C,I,O"    ( 停止監視 )

180 END
```

順便一提：刪除文字檔會同時將爲此檔案配給的檔案保留區與檔案在磁碟所佔用的磁區（ sectors ）釋放不再佔用。

3.3.2 CLOSE 指令——立即或延遲執行模式

在程式讀出或寫入檔案的工作結束之後，必須關閉檔案（ close file ）。適當地關閉每個檔案，可保證所有字元都寫入檔案，並且檔案保留區（ file buffer ）也被釋放供他用。

註：有關更多檔案保留區的用法，請參考 5.3 節。

註：檔案保留區(file buffer)，與另一譯名“檔案緩衝區”同義。

順便一提：程式應儘可能早一點將其開啓的檔案予以關閉，即使是暫時性也有必要。如果有需要，可以重新再開啓檔案。

有某些情況下，程式有誤，尚未關閉所有的開啓檔案之前，即已停止執行。當此種情況發生時，可從鍵盤打入CLOSE 指令。

CLOSE 指令採用以下的格式：

CLOSE [fn]

其中 fn 指示所欲關閉的檔案之名稱。沒有任何引數（argument）的CLOSE 關閉所有檔案。

程式技巧：可加入ONERR GOTO 指述於程式中，轉入一個副程式，將所有檔案關閉。

3.3.3 WRITE指令——延遲執行模式

WRITE 指令告訴DOS，欲寫入那個檔案，以及從何處開始寫入。WRITE 指令一直保持有效，直到下一個 DOS 指令或INPUT 指述出現爲止。您必須在PRINT 指述之前使用WRITE 指令，才能將字元寫入檔案。

在WRITE 指述之後，所有原先會顯示於螢幕上的輸出字元，全送到檔案。包括 INPUT 指述的問號提示字元（question -mark prompting character）（如果 INPUT 出現在WRITE 之後），及錯誤訊息（error message）（除非在ONERR GOTO指述後重新指定輸出裝置）。

指令格式如下：

WRITE fn [, Bn]

fn　　　指出所欲寫入資料的循序存取文字檔之檔名。此引數（

arguement) 是必要的。

[,Bn] 其中 n 是位元組 (bytes)或字元的數量，表示所欲跳過的字元數。此項目用來更改檔案的目前位址 (current position) 。

您可利用 Bn 選擇項，告訴 DOS 欲寫入檔案的第一個字元之位址。

順便一提： 如果您沒有採用ONERR GOTO指述，而 DOS 發出錯誤，則錯誤訊息 (error message) 存入您的檔案最後一欄。同時，錯誤訊息會取消WRITE 指令。

3.3.4 READ指令——延遲執行模式

READ指令告訴DOS，欲讀入資料的檔案，其有效期間直到下個 DOS 指令爲止。在使用 INPUT 和GET指述由檔案讀入資料之前，必須先有READ指述。

指令的格式如下：

READ fn [, Bn]

fn 指出意欲讀取資料的檔案之名稱，此引數是必要的。

[,Bn] n 是希望跳過的位元組或字元之數量，此選擇項目可變更檔案的目前位址 (current position)。

您可以利用 Bn 選擇項告訴DOS，要從檔案的那個位址讀取第一個字元。如果您不指定此項，DOS即從目前位址 (current position)讀取第一個字元。

注意： 當TRACE(Applesoft指述) 和 INPUT 指述併用時，您必須定義 D$ 變數的值含歸位字元 (carriage return) 和

CONTROL - D : D$=CHR$(13)+CHR$(4)

3.3.5 POSITION 指令——延遲執行模式

POSITION 指令使您可以存取檔案中任何資料欄的資訊。它會自動開啓您指定的檔案，並指示 DOS 從目前位址 (current position) 開始，越過數個資料欄才開始讀取資料。由於資料欄是一系列的字元，以歸位字元結束，POSITION 會一個位元組接一個位元組巡視檔案，尋找指定個數的歸位字元。

警告: POSITION 指令會取消READ和WRITE的指令；因此 POSITION 最好在READ或WRITE之前使用。

指令的格式如下

POSITION fn [, Rn]

fn 指定意欲改變目前位址的檔案之名稱，此引數是必要的。

[,Rn] 其中 n 是資料欄的個數，係指希望跳過的資料欄數，此數目相對於目前位址 (current position)。當 Rn 省略時，其預設値是 0 ，即第一欄。如果您指定的位址超出檔案結尾之外，DOS會顯示 END OF DATA 的訊息，並停止程式的執行。

本指令的目的是，將檔案的目前位址往前移動 n 個資料欄。當 R 値指定爲 0 時，DOS 從目前的資料欄開始讀取或寫入資料。如果 R 値是 1 ，則 DOS 跳過目前的資料欄，從下一個資料欄開始讀寫資料。

指定的位址係相對於目前位址。例如，假若目前位址是在檔案的第 4 資料欄，而您想讀第 10 欄，可利用 POSITION fn ,R6

跳過6個資料欄。

警告： 如果您利用POSITION 重寫目前的資料欄，須確定您輸出的字元數與原有的字元數相同。如果您輸出更少的字元，則會建立新的資料欄，此欄就在原資料欄的後半部。如果您輸出更長的字元，則會蓋掉下一個資料欄的一些字元。

3.3.6　APPEND指令——延遲執行模式

APPEND 指令開啓一個循序存取文字檔，並將目前位址（current position）設為檔案末尾，使您能加入資訊。開啓檔案之後，APPEND指令將目前位址指到檔案最後一個字元的下一個位元組。APPEND指令必須緊跟著WRITE 指令。

APPEND 指令的格式如下：

APPEND fn [, Sn] [, Dn] [, Vn]

fn　　指明欲增添資料的檔案，若檔案不存在，DOS會顯示 FILE NOT FOUND 的訊息。

[,Sn]　　其中n是由1至7的數目，指定所使用磁碟機其磁碟控制卡所在的擴充槽。如果此項省略，則 DOS 使用預設的槽號。

[,Dn]　　其中的n值是1或2，指定含有檔案 fn 的磁碟片所在的磁碟機。如果省略，DOS用預設的磁碟機號碼。

[,Vn]　　其中的n值由0至254，指定所欲使用的磁碟之卷號，如果您指定其為0，即只打V，或省略此項，則DOS不管卷號。

注意： 它不同於OPEN指令，APPEND 指令不會建立新的檔案，

因此所指定的檔案必須已存在。

下一個例子修改程式MAKE TEXT，使其能在文字檔之內加入數列資料。首先MAKE TEXT 讀入數列文字，存入陣列 A$，並詢問檔案名稱。然後開啓檔案，並輸出文字至檔案。

利用第一部磁碟機上的SAMPLE PROGRAMS磁碟，將MAKE TEXT載入記憶體，並將其顯示於螢幕上。

請打入如下的指令：

```
LOAD MAKE TEXT
LIST
```

修改程式，以便增添文字於檔案的末尾，將OPEN指述改成APPEND 指述。並將列 50 和列160 以如下的指述取代之：

```
50 PRINT "              TEXT FILE APPENDER"
160 PRINT D$;"APPEND ";N$
```

欲保存此程式，利用以下的指令：

```
SAVE APPEND TEXT
```

3.4 指令摘要

OPEN fn [, Sn] [, Dn] [, Vn]

開啓檔案，使 DOS 能讀出或寫入資料於此檔案。除非您的程式引用 POSITION 或 APPEND 指令，否則您必須在使用文字檔之前使用OPEN指令。若 fn 不存在，則 DOS 以此檔名建立一文字檔。若 fn 存在，則OPEN指令檢查此檔是否文字檔。

註: MAKE TEXT 係建立一循序存取文字檔的程式。

CLOSE [fn]

告訴DOS，您已結束讀寫檔案的工作。在程式結束之前，必須將所有已開啓的檔案關閉。

WRITE fn [, Bn]

告訴DOS，您要寫入那個檔案。在開啓之後，使用WRITE指令。DOS將資料寫入 fn 檔案，一直到其他 DOS 指令指定新的fn，才改變傳送資料的目的檔案。

READ fn [, BN]

告訴DOS，要從那個檔案讀取資料，以及從何處讀起。在您開啓檔案之後，使用READ指令。DOS從 fn 檔案讀取資料，直到有其他 DOS 指令指定新的 fn 才失效。

POSITION fn [, Rn]

設定檔案內的目前位址。POSITION 指令使您能跳過指定個數的資料欄，以便任意讀出或寫入檔案的其他資訊。

APPEND fn [, Sn] [, Dn] [, Vn]

開啓已存在的文字檔，並將目前位址前移到檔案末尾，使您的程式能加入資料於既存的檔案內。

注意: 只有CLOSE 能使用立即執行的方式（即由鍵盤輸入指令執行）。所有其他指令必須使用延遲執行模式，亦即由BASIC程式的PRINT 指述叫用。

4

隨意存取文字檔的使用

4.1 隨意存取文字檔 89

4.1.1 資料錄長度 90

4.1.2 寫入資料錄 91

4.1.3 讀出資料錄 92

4.2 示範：RANDOM 程式 92

4.3 隨意存取的樣本程式示例 94

4.3.1 程式的執行次序 94

4.3.2 儲存資料錄 95

4.3.3 寫入資料錄 96

4.3.4 讀出資料錄 98

4.4 隨意存取文字檔的指令 100

4.4.1 OPEN指令—延遲執行模式 100

4.4.2 CLOSE指令—立即執行或延遲執行模式 102

4.4.3 WRITE 指令—延遲執行模式 102

4.4.4 READ 指令—延遲執行模式 104

4.5 指令摘要 105

第4章

隨意存取文字檔的使用

本章討論隨意存取文字檔（random-access text files）：如何建構此類檔案及如何使用，同時也包含兩份示範程式，本章最後討論隨意存取指令（random-access command）及其語法。

注意：循序存取文字檔（sequential-access text file）是一連串不定長的資料欄（field）。隨意存取文字檔，則是一連串定長的資料錄（record），每錄可能含數個資料欄。

4.1 隨意存取文字檔

隨意存取文字檔（random-access text file）就好像一本沒有限制頁數（即資料錄個數）的筆記本。每頁可保存相同數目的字元。但是這些字元可分散在不限數目的數列文字（即資料錄中的資料欄）。使用隨意存取文字檔（即筆記本）時，您在讀寫資料欄（每列文字）之前，必須先跳至某一特定資料錄（即頁）。

要使用隨意存取指令（random-access command）之前，您必須：

- 在開啓檔案時，指定資料錄長度（record length）。

- 當您要讀寫資料時，須指明資料錄的編號。

好主意：如欲有效使用您的隨意存取文字檔，必須牢記住其相關的資訊：結構（structure）、格式（format）、資料錄長度（record length）和資料欄長度（field length）。可將這些資訊，加於各程式開頭的註解中，例如：

```
10 REM R/A FILE; RECORD LENGTH 5; ONE FIELD PER
     RECORD
```

最好，將這些資訊也引入檔案名稱中，例如，

```
R/A FILE-L5F1
```

4.1.1 資料錄長度

當您開啓隨意存取文字檔時，需指定**資料錄長度（record length）**，即每個資料錄所存的字元數。例如，開啓一隨意存取文字檔，檔名爲STOCK SYMBOLS，利用以下指令：

```
30 PRINT D$;"OPEN STOCK SYMBOLS,L12"
```

其中參數L指明檔案STOCK SYMBOLS中各資料錄長度是12個字元。

您也可以用 16 進位數字來指定資料錄長度，例如：

```
30 PRINT D$;"OPEN STOCK SYMBOLS,L$C"
```

警告：如果資料錄長度存於一個變數，且此長度值小於 10 時，千

萬勿以此變數來指定資料錄長度。

順便一提：當您開啓隨意存取文字檔，無須指明檔案中總共的資料錄個數，DOS 爲您做這件事。

4.1.2　寫入資料錄

利用 WRITE 指令來使用隨意存取文字檔時，必須指明您所要寫入的資料錄之編號。如果此號碼的資料錄並不存在，DOS 建立此錄，並爲此資料錄預留足夠的磁碟空間。
例如，欲輸入資料到 STOCK SYMBOLS 檔案的 10 號資料錄，可利用以下的指令寫入資料。

```
50 PRINT D$;"WRITE STOCK SYMBOLS,R10"
```

順便一提：也可以用 16 進位數字來指定資料錄，例如

```
50 PRINT D$;"WRITE STOCK SYMBOLS,R$A"
```

然後利用 PRINT 指述將字元放入 STOCK SYMBOLS 檔案的 10 號資料錄。即使您只輸入一個字元，資料錄仍然佔用OPEN 指述所指定的長度。

警告：請注意，不得輸入超出資料錄範圍的字元，DOS 會將多餘的字元放入下一個資料錄。記住，在資料欄末尾的 carriage return 也算一個字元。

註：資料錄長度即是每個資料錄所允許的字元數。

4.1.3 讀出資料錄

當您利用READ指令來使用隨意存取文字檔，必須指定所要讀取的資料錄之編號。例如，欲讀取STOCK SYMBOLS檔案的第七錄，則打入

```
90 PRINT D$;"READ STOCK SYMBOLS,R6"
```

然後利用INPUT或GET指述來讀取，如果所要讀的資料錄不存在，您看看到 END OF DATA 的訊息。

4.2 示範：RANDOM 程式

RANDOM程式存於 SAMPLE PROGRAMS 磁碟片，利用隨機存取文字檔來示範庫存管理計劃。RANDOM是一非常小的程式：最多可處理 9 種物料。任何Apple II電腦都能處理上千種物料的庫存資料。

RANDOM自動執行隨意存取文字檔 APPLE PROMS。您可以列出庫存中的一項或所有項目，也可以更改項目，能一次改一項或同時改所有項目，其步驟如下：

磁碟片 SAMPLE PROGRAMS 放入磁碟機，打入

```
RUN RANDOM
```

您將看到 3 個選擇項(option)：

註：檔案的第一個資料錄其編號為 0 。

```
        APPLE PROMS
COMMAND          NUMBER
-------          ------
LIST                1
CHANGE              2
EXIT                3
CHOOSE NUMBER (1 - 3) 1
```

記住：每次回答後，要按下(RETURN)鍵。

按下1鍵，會出現以下的訊息：

```
PART NUMBER 1-9 (0=ALL) 0
```

按下0鍵，即可得到庫存的所有物料之清單，畫面如下：

```
PART#  NAME               SIZE  IN STOCK
-----  ----               ----  --------
  1    PARALLEL PRINT     256   500
  2    COMMUNICATIONS     256   1250
  3    (NOT AVAILABLE)    256   0
  4    (NOT AVAILABLE)    256   0
  5    DISK BOOT          256   432
  6    STATE MACHINE      256   460
  7    SERIAL PRINTER1    256   878
  8    SERIAL PRINTER2    512   741
  9    CENTRONICS         256   1290

PRESS THE RETURN KEY TO CONTINUE
```

如果您想回到選擇項，請按(RETURN)鍵。

試一試不同的選擇項。如果您選擇第一項，則可以依據物料號碼列出各物料，也可以同時列出所有物料。

第2選擇項則讓您更改所有物料名稱及其說明。例如，假設物料3號是COSMIC GLUE，尺寸56，存量1234。如欲修改物料3號，則挑第2選擇項（CHANGE，更改），然後選物料3號，舊的物料名稱會顯示出來，同時游標停駐於開頭處，讓您輸入新名稱。當您打入名稱並按下(RETURN)鍵，游標會往右移，以同樣的方式

進行修改物料尺寸和數量。如果要保留目前的名稱、尺寸或數量，則只需按(RETURN)鍵即可。

第 3 選擇項停止程式執行。

4.3 隨意存取的樣本程式示例

SAMPLE PROGRAMS 磁碟片中的ADDRESS程式，可說明隨意存取文字檔如何使用。ADDRESS程式用以維護郵購名單，主要有兩項工作：加入新地址以及尋找已輸入的地址。每項工作各以一副程式來執行。

4.3.1 程式的執行次序

主程式十分簡單淺顯，如下所列：

```
10 D$ = CHR$ (4)              ( P$是CONTROL-D )
20 F$ = "BLACK.BOOK"          ( F$存地址檔案之名稱 )
30 HOME                       ( 清除畫面 )
40 PRINT "WHAT WOULD YOU LIKE TO DO?"
50 PRINT " 1 ENTER A NEW ADDRESS"
60 PRINT " 2 LOOK UP AN ADDRESS"
70 PRINT " 3 END"
80 INPUT "TYPE A NUMBER AND PRESS RETURN ";C$
90 C = VAL (C$)               ( 將輸入數字換成整數 )
100 IF C < 1 OR C > 3 THEN GOTO 30
                              ( 不正確的輸入，再問一次 )
110 IF C = 1 THEN GOSUB 1000
                              ( 輸入新地址 )
120 IF C = 2 THEN GOSUB 2000
                              ( 尋找地址 )
130 IF C = 3 THEN END
140 GOTO 30
```

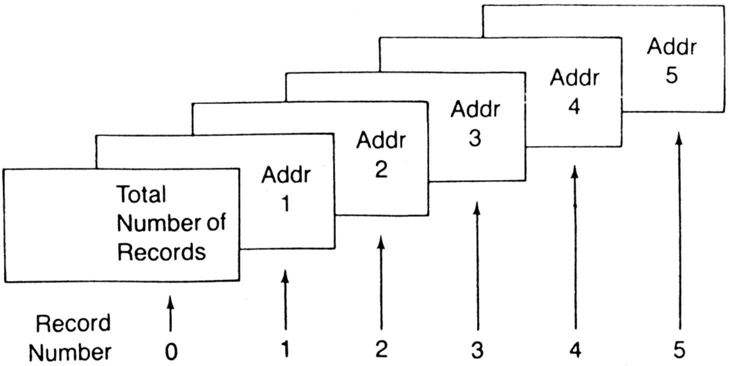
列 80 至列 100 的工作是，讓您選擇，要輸入資料或是要讀出已有的資料，還是跳出程式。您打入一字母，挑中某一選擇項，則 VAL 指述（即列 90 ）將其轉換成數目。列 100 指示程式重覆詢問的工作（列 40 ），以應付輸入數字超出範圍的情況。列 110 至 130 分別將程式轉入您所挑中選擇項所對應的副程式。

4.3.2 儲存資料錄

當您建立隨意存取文字檔時，必須決定如何安排各資料錄的資訊，以及資料錄有多長。例如，ADDRESS 程式將資料存於 BLACK. BOOK 資料檔。BLACK. BOOK 的各資料錄佔用 200 個字元，包含 6 個資料欄：姓名、地址、城市、州、郵遞區號和電話號碼。

ADDRESS 將各資料欄分別存入各個變數（N$, A$, C$, S$, Z$ 和 P$），然後將資料錄總數（TR）加 1 。ADDRESS 程式將第一個地址存入第 1 錄，第二個存入第 2 錄，餘此類推。第 0 錄則儲存檔案的資料錄總數。圖 4-1 顯示檔案 BLACK. BOOK 的結構，其中假設此檔含有 5 個地址。

圖 4-1　5 份地址存入 BLACK. BOOK

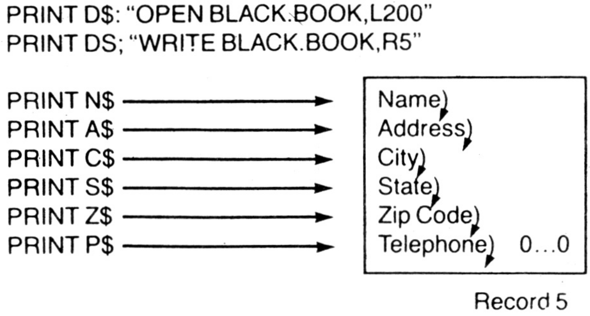


4.3.3 寫入資料錄

在寫入新的資料錄之前，ADDRESS首先利用OPEN和WRITE 指令，指定BLACK. BOOK要接收新的資料錄。然後各列PRINT 指述依次將各資料欄填入資料錄，並將新的總數填入第 0 錄。

圖 4-2說明 BASIC 指述如何將地址放入BLACK. BOOK的第 5 錄。其中 ↲ 符號表示carriage return字元，0……0 則表示填入ASCII 0 以便將資料錄填滿 200 個字元。

圖 4-2　將地址寫入第 5 錄



副程式READ NEW INFO中列1010至列1060的BASIC指述，係將一資料錄的資訊加以收集保存。

```
1000 REM READ NEW INFO
1010 INPUT "NAME:        ";N$
                          （讀取欲輸入地址之姓名）
1020 INPUT "ADDRESS:     ";A$
                          （讀取地址）
1030 INPUT "CITY:        ";C$
                          （讀取城市）
1040 INPUT "STATE:       ";S$
                          （讀取州名）
```

```
1050 INPUT "ZIP CODE:  ";Z$
                          （讀取郵遞區號）
1060 INPUT "PHONE:      ";P$
                          （讀取電話號碼）
```

列 1070 至列 1120 開啓檔案，檔案名稱儲存於變數 F$ 中，將資訊寫入新資料錄（資料錄數量是TR）。

```
1070 PRINT D$;"OPEN ";F$;",L200"
                          （開啓檔案指定錄長爲
                           200）
1080 PRINT D$;"READ ";F$;",R0"
                          （讀取總計資料錄）
1090 INPUT TR:TR = TR + 1
                          （擷取資料錄總數；並將其加 1）
1100 PRINT D$;"WRITE ";F$;",R";TR
                          （準備寫入第TR資料錄）

1110 PRINT N$: PRINT A$:   PRINT C$
                          （將地址資料各部份分別存入各欄）
1120 PRINT S$: PRINT Z$:   PRINT P$
```

列 1130 至列 1160 是將檔案中新的地址資料錄之總數，寫入第 0 錄，關閉檔案，然後結束此副程式。

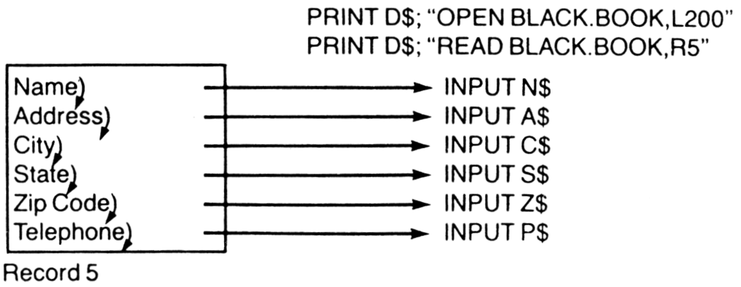
```
1130 PRINT D$;"WRITE ";F$;",R0"
                          （準備寫檔案第 0 錄）

1140 PRINT TR             （寫入新的資料錄總數）
1150 PRINT D$;"CLOSE ";F$
                          （關閉檔案）
1160 RETURN               （結束副程式）
```

4.3.4 讀出資料錄

瞭解如何將資訊存入資料錄之後，即可決定如何擷取您的隨意存取檔中某一特定資料錄。副程式LOOKUP提供解決的方法：利用OPEN和READ指令指定BLACK.BOOK檔案爲讀取資料的檔案之後，LOOKUP首先從BLACK. BOOK中讀取資料錄總數（第0錄）。然後LOOKUP由各資料錄中擷取姓名，並將其顯示於一張編號的名單上。最後，LOOKUP 詢問您要顯示那一個資料錄，並將您所選的顯示出來。

圖 4-3 說明檔案BLACK.BOOK第5錄的資料欄相關的BASIC指述其執行情況。

圖 4-3　由第 5 錄讀取一地址資料



列2000至列2030係由檔案的第0錄讀取地址資料之總數。列2050至列2100則讀取各資料錄的姓名，並將其顯示於螢幕上。

```
2000 PRINT D$;"OPEN ";F$;",L200"
                              （開啓檔案）
2010 PRINT D$;"READ ";F$;",R0"
                              （讀第0錄）
2020 INPUT TR                 （擷取資料錄總數）
2030 IF TR = 0 THEN GOTO 2210
                              （檢查有否資料錄存在）
```

```
2040 HOME                  ( 清除畫面 )
2050 PRINT "WHOSE ADDRESS DO YOU WANT?"
2060 FOR I = 1 TO TR       ( 對於每個資料錄 I ,
2070  PRINT D$;"READ ";F$;",R";I
                            定位至第 1 錄
2080  INPUT N$              並讀入所存的姓名 )
2090  PRINT I,N$           ( 顯示錄號和姓名於螢幕 )
2100 NEXT I                ( 重覆執行各資料錄 )
2110 PRINT D$              ( 空的 DOS 指令，關掉上一個
                             READ 指令 )
```

列 2010 的 READ 指述其目的是指定由檔案 BLACK. BOOK 中讀取字元，而不是由鍵盤讀資料。如欲從鍵盤讀取資料，則必須將列 2010 去掉。列 2110 的 DOS 指令只輸出 CONTROL -D，亦即是將檔案的讀取動作關掉。

列 2120 請求輸入一個號碼，將其讀入字串變數 R$，列 2130 將 R$ 轉換成數字存入 R。如果您按入任何字母鍵，則函數 VAL 將其轉換成 0。再下一列的指述將輸入的數字與有效位址數相比較，如果是不適當的值，則必須重新顯示問題一次，讓您再次輸入正確值（列 2140）。

```
2120 INPUT "TYPE A NUMBER AND PRESS RETURN";R$
2130 R = VAL (R$)          ( 讀取回答的數字 )
2140 IF R < 1 OR R > TR THEN GOTO 2120
                           ( 如果不正確的數目，再試一次 )
```

LOOKUP 副程式的最後一部份可顯示地址，列 2160 的 READ 指述設定所要的資料錄之位址。列 2170 的 INPUT 指述讀出此錄的 6 個資料欄，列 2180 至列 2200 將此 6 個資料欄顯示於螢幕上。列 2220 用來防止在您讀入地址資料之前，地址就已從畫面中消失。

```
2150 HOME                     (清除畫面)
2160 PRINT D$;"READ ";F$;",R";R
                              (準備讀第R錄)
2170 INPUT N$,A$,C$,S$,Z$,P$
                              (讀入地址資料)
2180 PRINT N$: PRINT A$       (顯示姓名地址)
2190 PRINT C$, S$             (城市名和省名顯示於同一行)
2200 PRINT Z$: PRINT P$       (顯示郵遞區號和電話號碼)
2210 PRINT D$;"CLOSE ";F$
                              (關閉檔案)
2220 VTAB 23: HTAB 10         (設定游標位址,保持原畫面)
2230 INPUT "PRESS RETURN TO CONTINUE";T$
2240 RETURN
```

4.4 隨意存取文字檔的指令

除了CLOSE之外，其他用於隨意存取文字檔(random-access text file)的指令只適用於延遲執行的方式。

仔細研讀：在此所描述的指令，與循序存取文字檔的指令略有不同，它們具有不同的選擇項。

4.4.1 OPEN指令—延遲執行模式

在程式讀寫一隨意存取文字檔之前，必須利用OPEN指令開啓(open)此文字檔。指令的一般格式是，

註：循序存取文字檔(Sequential-access text file)，請參考第3章

OPEN fn, Ln [, Sn] [, Dn] [, Vn]

fn　　是所要開啓檔案的名稱，以便使用隨意存取檔案的資料。如果 fn 存在，DOS 會檢查是否爲文字檔。如果檔案已存在且已經開啓，DOS 會將指明的檔案關閉（close），然後再開啓。如果檔案並不存在，DOS 以此檔名建立（create）一文字檔，並開啓此檔。

Ln　　其中 n 是從 1 至 32767 的任一數字，指定檔案中各資料錄的位元組（byte）數。無論何時開啓檔案，亦即無論是建檔或重新開啓，都必須指定資料錄長度（record length）。如果您省略 L 項。DOS 假設其值爲 1 。

[,Sn] 其中 n 是 1 至 7 的任一數字，指定所欲指用磁碟機之磁碟控制卡所插的擴充槽編號。如果省略此項，則DOS採用預設的槽號（default slot number）。

[,Dn] 其中 n 是 1 或 2 ，指定存有檔案 fn 之磁碟所在的磁碟機。如果省略此項，DOS採用預設的磁碟機號碼（default drive number）。

[,Vn] 其中 n 是 0 至 254 的任一數字，指定所用磁碟片的卷號。如果您指定爲 0 ，或只打 V ，或省略此選擇項，則DOS不管卷號（volume number）。

當您第一次開啓某一隨意存取文字檔時，即以指定的資料錄長度建立此檔。爾後每次開啓此檔時，必須指定相同的長度。

警告：宣告的資料錄長度必須大於您所要放入檔案資料錄的可能最大字元數，包括 carriage return 字元。如果您寫入的資料超出資料錄的末尾，將會寫到檔案下一個資料錄的開頭，而毁掉原先該處的資料。

每當任何文字檔開啓後，DOS準備從檔案起頭開始讀寫，同時指定一檔案保留區（file buffer）存放檔案有關的資訊。

4.4.2 CLOSE指令—立即執行或延遲執行模式

在讀寫資料於檔案之後，程式必須關閉此檔案，以保證 DOS 寫入所有的字元，並將檔案相關的保留區釋放。

CLOSE的格式如下：

CLOSE [fn]

其中 fn 指定所要關閉的檔案之檔名。如果您省略檔案名稱，則CLOSE 指令關閉所有開啓的檔案，結束程式的任何寫入的動作，並釋放所有的檔案保留區（file buffer）。

有時程式執行有錯誤，可能在關閉所有已開啓的檔案之前停止執行。發生此種情況之後，可從鍵盤打入CLOSE 指令，將任何已開啓的檔案關閉。

程式技巧：在程式中加入ONERR GOTO指述，假若錯誤發生時即可轉入一副程式，將任何已開啓的檔案關閉。

4.4.3 WRITE指令—延遲執行模式

WRITE 指令表明您欲寫入資料至那一個隨意存取檔，指定將接受資料的資料錄編號，同時指明欲寫入的第一個字元是從資料錄中的那個位址開始的。

註：檔案保留區（File buffer）=595 位元組（bytes）

WRITE 指令格式如下：

WRITE fn [, Rn] [, Bn]

fn　　指出要寫入資料的檔案之檔名。

[,Rn]　其中 n 由 0 至 32767 的任一數字，指定程式要寫入資料的資料錄號。如果 Rn 省略，DOS 準備從第 0 錄寫入。

[,Bn]　其中 n 是DOS在指定的資料錄中要越過的字元數，更改檔案中的目前位址（current position）。如果此選擇項省略，則DOS 準備從第 0 個位元組開始寫入資料，即 R 項所指定資料錄的開頭。

WRITE的有效期限直到下一個DOS指令，或下一個 INPUT 或 GET 指述出現爲止。

在使用INPUT 指述將字元放入資料錄之前，必須使用WRITE 指述；假若您要寫入的資料錄不是目前所在的位址時，就必須使用 WRITE 指令。

警告：如果您沒有利用ONERR GOTO指述，而且您的程式產生錯誤訊息，則此“訊息”會寫入您的文字檔內最後一欄，同時錯誤訊息會取消WRITE 指令。

WRITE 指述用來告訴DOS，將所有平常要送到螢幕上的字元，全部送到開啓的檔案。也包括錯誤訊息（如果您沒有引用ONERR GOTO 指述）與 INPUT 指示顯示的提示字元（如果 INPUT 指述出現於 WRITE 指令之後）。

警告：千萬勿使用循序存取指令 POSITION，來跳過資料錄的某些資料欄。如果您這樣做，將會取消WRITE 指令。

您所輸出的字元總數勿超出OPEN的 L 引數所指定的字元數（包括 carriage return 字元）。DOS 會正確地寫入目前

的資料錄，也會計算下一個資料錄的起始位址，好像上一個資料錄具有指定的長度。下一個資料錄寫入資料時，會蓋去上個資料錄過長的部份，包括標定資料錄結尾的 carriage return 字元。

如果您輸入"隨意存取資料錄"的字元數少於OPEN的L引數所指定的數目，將會建立兩個資料欄：您所輸出的資料欄和緊接其後的"空"資料欄。

4.4.4 READ指令—延遲執行模式

當您使用隨意存取檔時，由READ指令指定爾後的 INPUT 和 GET 指述希望讀取字元的檔案。您可以指定檔案中的資料錄編號（Rn）和資料錄中的位元組位址（Bn），資料即自指定的位置開始被讀取出來。

READ指令的有效期限直到下一個 DOS 指令出現爲止；其格式如下：

READ fn [, Rn] [, Bn]

fn　　指定所要讀取資料的檔案之檔名。

[,Rn]　其中 n 是 0 至 32767 的任一數目，指定程式由那一錄讀取字元。如果 Rn 省略，DOS 準備從第 0 資料錄開始讀取資料。如果 Rn 指定的資料錄並不存在，而您又沒有使用ON-ERR GOTO指述，則第一次出現的 INPUT 或 GET 在讀取不存在的資料錄之後，立即出現 END OF DATA 訊息。

[,Bn]　其中 n 是 DOS 在指定資料錄中跳過的字元數，會改變檔案的指標移到相對於目前位址（current position）n 個字元。

注意：除非D$=CHR$(13)+CHR$(4) 已使用於程式中，否則避免將Applesoft 的TRACE指令和INPUT 指述合用一您的程式會停止執行，並等著從鍵盤接受輸入資料。

4.5　指令摘要

OPEN fn, Ln [, Sn] [, Dn] [, Vn]

開啓〃隨意存取文字檔〃，使DOS能寫入資料。長度項(L, length)指定資料錄的位元組或字元數。如果檔案係新建，即開啓前該檔案尚未存在，則DOS根據指定的檔名 fn 建立一檔案。如果 fn 存在，DOS 檢查其是否為文字檔。

CLOSE [fn]

告訴DOS您已結束讀寫的工作, 進行關閉檔案的動作。如果fn省略，DOS 關閉所有已開啓的檔案。

WRITE fn [, Rn] [, Bn]

指定要寫入資料的隨意存取檔，以及開始寫入資料的位址：資料錄編號 (Rn) 及寫入資料錄第一個字元的位址(Bn)。檔案必須已開啓。DOS 將資料寫入 fn, 直到下一個DOS指令或下一個 INPUT 或GET 指述出現才失效。

READ fn [, Rn] [, Bn]

指定所要讀取的隨意存取檔和開始讀取資料的位址：資料錄編號(Rn)和從資料錄中讀取的第一個字元之位址(Bn)。檔案必須已開啓。DOS 從檔案 fn 讀取資料，直到下個DOS指令出現為止。

記住：上述所有指令（包括 CLOSE ）都可在 BASIC 程式中使用，但是只有CLOSE 可以從鍵盤下命令。

5 高級程式技巧

5.1 指令檔之使用 109

5.1.1 EXEC示範 110

5.1.2 建立指令檔 112

5.1.3 特殊用途的指令檔 115

5.1.3.1 從BASIC 程式抽取指述列 115

5.1.3.2 將機器語言翻譯成BASIC程式 116

5.2 EXEC指令 118

5.3 MAXFILES 指令 120

5.3.1 保留區的資訊 120

5.3.2 範例 122

5.4 CHAIN 122

5.4.1 CHAIN 指令— Integer BASIC 122

5.4.2 CHAIN 程式—Applesoft 125

5.5 MASTER 程式 126

5.5.1 範例 127

5.6 製作轉鑰式磁碟 129

5.6.1 建立轉鑰式磁碟 129

5.6.2 建立主轉鑰式系統 130

5.6.3 萬用磁碟 130

5.6.3.1 範例 131

5.7 摘要 134

5.7.1 指令 134

5.7.2 程式 135

第5章
高級程式技巧

本章描述 3 個指令和 2 份程式用法，使得 **DOS** 得以和數個程式、檔案、系統交互作用。

- **EXEC**指令讓您利用指令檔（command file）來執行其他的檔案。
- **CHAIN** 程式讓您將兩份程式連結在一起。
- **MAXFILES** 指令讓您增加或減少在您的**Apple II**記憶體中“同時開啓”的檔案。
- **MASTER** 程式讓您建立主磁碟（master disk），使其適合**Apple II**電腦不同容量的記憶體。

5.1 指令檔之使用

Apple II電腦可以從特定的文字檔中讀取指令代替從鍵盤讀取資料，此種檔案稱做**指令檔**（**command file**）。指令檔讓您可以自動執行常用的指令集，可以將某一種 **BASIC** 語言轉換成另一種，或者自動輸入數個程式常用的同一份資料。要使用此程式功能，首先要寫一個 **BASIC** 程式建立指令檔，然後利用**EXEC**指令執行指

令檔。

指令檔內可包含所有可以從鍵盤輸入的指令 DOS指令、BASIC 指述、甚至於監督程式之指令。當您將 DOS 指令放入指令檔時，無須在前頭加上 CONTROL - D 。

您無法利用LIST 指述來看指令檔的內容，必須操作SAMPLE PROGRAMS磁碟片上的程式GET TEXT，或者特殊的編輯程式，如Apple Write 2.0或EDSAM。您也可以利用MON C指令，在指令檔執行內含指令時，查看這些指令。

以下的示範是利用GET TEXT來看看指令檔DO'ER內容。

記住: 指令檔中的 DOS 指令無須在前頭加上 CONTROL - D。

5.1.1 EXEC示範

此示範分成兩部份，首先執行SAMPLE PROGRAMS磁碟片中的 BASIC 程式EXEC DEMO，以建立指令檔。然後下達EXEC指令，您的電腦就從指令檔讀取指令而不由鍵盤。

順便一提: 確定您的SAMPLE PROGRAMS磁碟可寫入資料，以便存入指令檔。

1. 要建立指令檔，將您的SAMPLE PROGRAMS磁碟插入第一部磁碟機，並輸入

```
RUN EXEC DEMO, D1
```

而後，顯示如圖5-1的畫面，請仔細閱讀文中的說明。

註: 指令檔(command file) = EXEC 檔。
指令檔的檔案型態(file type)是T。

註: 有關EDSAM請參考Applesoft DOS Tool Kit。

```
          << EXEC DEMO >>

THIS PROGRAM CREATES A SEQUENTIAL TEXT
FILE NAMED "DO'ER" CONTAINING SEVERAL
STRINGS, EACH A LEGAL APPLE II COMMAND.

THEN YOU TYPE:
EXEC DO'ER
THEN THE COMMANDS IN FILE DO'ER TAKE
CONTROL OF YOUR COMPUTER. EACH COMMAND
WILL BE EXECUTED JUST AS IF IT HAD BEEN
TYPED AT THE KEYBOARD. THE DOS MANUAL
DESCRIBES THE PROGRAM IN MORE DETAIL.

        << HAPPY EXECUTING >>

PRESS THE SPACE BAR TO MAKE THIS
PROGRAM CREATE THE FILE DO'ER.

IF YOU WISH TO STOP THIS PROGRAM NOW,
YOU MAY PRESS THE ESC KEY.
```

圖 5-1 EXEC示範的第一個畫面

2. 閱讀EXEC DEMO的第一個畫面說明，按下 SPACE 鍵。EXEC DEMO 將DO'ER 檔案寫入磁碟，然後畫面顯示如圖 5-2。

```
IT'S DONE!!!

YOUR APPLE'S READY TO DO'ER ITS THING!
ALL YOU HAVE TO DO IS TYPE
            EXEC DO'ER
PRESS THE RETURN KEY, AND SIT BACK.
```

圖 5-2 EXEC DEMO的最後畫面

EXEC DEMO已建立一指令檔，檔名是DO'ER ，並將其存入磁碟。

3. 要了解指令檔的動作，執行DO'ER ，請輸入 EXEC DO'ER

DOS 從磁碟讀入 DO'ER 的第 1 磁區，並開始執行指令 DO'ER 在螢幕上同時描述所有它正在做的事項。

由於 DO'ER 執行某些程式會用到 Integer BASIC 語言系統，如果您的電腦沒有此種語言，則會出現 LANGUAGE NOT AVAILABLE 的字樣。不必擔心，DO'ER 只執行它能做到的程式。而且您無須按任何鍵，只有當您的檔案目錄超出 18 列時，必須按 RETURN 鍵才會出現第 19 列之後的檔案名稱。

4. 要查看 DO'ER 檔案中的指令，可利用 GET TEXT 程式。請打入

```
RUN GET TEXT
```

當 GET TEXT 詢問 LIST WHAT TEXT FILE? 時，打入

```
DO'ER
```

GET TEXT 顯示 DO'ER 的內容時，仔細看看它所包羅的各式各樣指令。

5.1.2 建立指令檔

要建立指令檔，此 BASIC 程式必須：

- 利用 OPEN 指令開啓文字檔。
- 利用 WRITE 指令準備所要寫入的檔案。
- 利用 PRINT 或 LIST 將指令放入文字檔。
- 關閉文字檔。

以下是逐步漸進的例子，說明如何建立EXEC檔。圖 5-3 說明此例子所用的指令和檔案。本例所用的檔案有MAKE.DOIT 程式（一個建立指令檔的 BASIC 程式）和一個DOIT指令檔（用來執行AWAY程式）。DOIT 包含以下的 DOS 指令 :CATALOG、RUN和 LIST（ BASIC的指令）。

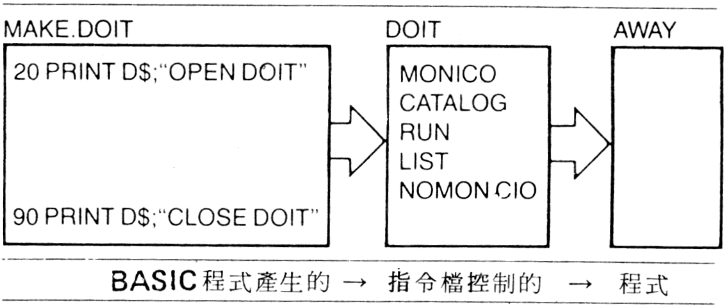


圖 5-3　指令檔的例子

1. 首先打入以下的程式：

```
5   REM MAKE.DOIT
10  D$ = CHR$ (4)          （D$是CONTROL-D）
20  PRINT D$;"OPEN DOIT"
                            （準備檔案的開啓及寫入）
30  PRINT D$;"WRITE DOIT"
40  PRINT "MONICO"          （監視操作）
50  PRINT "CATALOG"         （將這三個指令放入指令檔
60  PRINT "RUN AWAY"        DOIT）
70  PRINT "LIST"
80  PRINT "NOMON CIO"       （解除監視功能）
90  PRINT D$;"CLOSE DOIT"（關閉檔案）
```

將此程式保存（save）起來，取名爲MAKE.DOIT，打入

```
SAVE MAKE.DOIT
```

PRINT 指述中的字串以 D$ 開頭者（列 20,30 和 90)是

DOS 指令，當您執行 MAKE.DOIT 時，會執行這些指令。其他的 PRINT 指述則會寫入資料至指令檔 DOIT，供爾後利用。注意，DOS 指令放入指令檔時，無須在前頭附上 CONTROL - D 。

2. 接著打入此 Applesoft 程式：

```
5 REM AWAY
10 PRINT "A WAY TO JOURNEY,"
20 PRINT "A WHALER JOE."
30 PRINT "AWEIGH THE ANCHOR,"
40 PRINT "AWAY WE GO."
```

保存此程式到磁碟，打入

```
SAVE AWAY
```

爾後，指令檔會執行AWAY程式。

3. 當您保存MAKE. DOIT和AWAY程式之後，執行MAKE. DOIT 程式以建立循序存取文字檔DOIT，打入

```
RUN MAKE.DOIT
```

當 MAKE. DOIT 結束時，DOIT 檔案已在磁碟上。

4. 可看看這些指令執行的程序，其過程有如各種指令快速地由鍵盤輸入，並一一地執行。執行EXEC檔案內的指令，請打入

```
MON I,C,O
EXEC DOI
```

DOIT 指令檔顯示磁碟中的目錄，印出一些句子以及列出 AWAY本身程式。

注意： 您也可以利用程式MAKE TEXT來建指令檔，但是無法利用MAKE TEXT程式來修正已存在於指令檔內的錯誤。

5.1.3　特殊用途的指令檔

您可以設計指令檔來執行很多工作；將 BASIC 程式的列印結果轉換成文字檔，或使用與 DOS 互通的文書處理程式來編輯程式，將程式某一部份放入其他程式，或將副程式當中的副程式插入另一程式中，或連接兩個程式。以下的例子說明這兩類的工作。

5.1.3.1 從BASIC程式抽取指述列

EXEC指令讓您從記憶體中的程式抽取幾列，並將其插入文字檔以建立新程式。抽取的數列指述必須加以編輯（edit）才能執行此新程式。也許有點麻煩，但是編輯已有的指述畢竟比重新輸入來得容易些。

下一個例子是CAPTURE程式，此程式從記憶體中讀入數列 BASIC 程式，並將其放入文字檔 LISTING 。

```
1 REM CAPTURE ROUTINE
2 D$ = CHR$ (4) : POKE 33,33
                              (D是CONTROL-D;更
                              改文字區域，以防止PRINT
                              指述的空白。)
3 PRINT D$;"OPEN LISTING"
                              (準備寫檔案LISTING)

4 PRINT D$; "WRITE LISTING"
5 LIST 2270,5130              (列印要寫入檔案的指述列)
6 PRINT D$; "CLOSE LISTING"(關閉檔案)

7 TEXT : END                  (重設文字區域)
```

列 2 的POKE指述將一數值放入記憶位址，POKE 33,33設定文字區的橫列距離是 33 個字元（左、右邊界相距 33 格），使

註：MAKE TEXT 可參考第 3 章。

BASIC 不至將過長的一列分成兩橫列且 DOS 不會加入多餘的"空白字元"。

利用 EXEC LISTING 指令將 LISTING 的各列讀入電腦記憶體，就好像您已輸入這些指述。

您可以在程式的開頭加入 CAPTURE 副程式，然後利用它來剪裁此程式。將列 5 中的列號以您所要從記憶體中抽取之列號加以取代；將檔名 LISTING （列 3 、 4 和 6 ）改名，產生另一檔案，抽取的數列指述即存入此檔。

您也可以利用 CAPTURE 副程式來組合程式。首先利用CAPTURE建立一程式，然後將第 2 個程式載入記憶體，利用EXEC 指令執行 CAPTURE 所建的檔案，將第 1 個程式載入記憶體。

記住：使用EXEC執行某一檔案，不會刪去已在記憶體的程式。

5.1.3.2 將機器語言翻譯成BASIC程式

SAMPLE PROGRAMS磁碟片上的示範程式POKER，可建立一 EXEC檔案，用來讀取二進制的機器碼程式（binary program），並將其翻譯成 BASIC 程式。POKER 程式利用PEEK指述來讀取二進制機器碼程式中連續的位元組（byte），並且對於每個位元組都產生POKE指述並寫入一個指令檔內。

當您執行此指令檔時，會建立一新的 BASIC 程式，其中含有這些POKE指令。您也可以改寫此程式，使得它所產生的指令檔，可以將新的 BASIC 指述加入記憶體中已有的 BASIC 程式內。

註：PEEK取得記憶體中某個位址所存的值；POKE則將值直接放入指定的記憶位址。

```
100 REM PROGRAM POKER      （D$是CONTROL-D）
110 D$ = CHR$ (4)
120 HOME
130 INPUT "FILE NAME TO CREATE? ",N$
140 PRINT : INPUT "STARTING ADDRESS (DECIMAL)?
      ";S
150 PRINT : INPUT "ENDING ADDRESS (DECIMAL)?
      ";E
160 IF S>E THEN PRINT : PRINT "THE STARTING
      ADDRESS MUST BE LESS THAN" : PRINT "THE
      ENDING ADDRESS"; CHR$(7): GOTO 140
170 PRINT : INPUT "BASIC LINE  TO START AT?
      ";LINE
180 IF LINE<0 OR LINE>63999 THEN PRINT : PRINT
      "BASIC LINE NUMBERS MUST BE IN THE
      RANGE": PRINT "OF 0 TO 63999"; CHR$(7):
      GOTO 170
190 PRINT D$;"OPEN "N$      （列 130所指名的檔案如果
200 PRINT D$;"CLOSE "       已存在，則將其開啓、關閉、
210 PRINT D$;"DELETE "N$    、刪除）
220 PRINT D$;"OPEN "N$      （開啓列130所指名的檔案）
230 PRINT D$;"WRITE "N$     （準備寫入資料）
240 FOR PLACE = S TO E      （對於每個記憶位址計值增
250 C = C + 1               加1，每列放入10個
260 IF C = 10 THEN C = 1    POKE）
270 IF C < > 1 THEN 300     （對於各列第一個POKF，輸
280 PRINT : PRINT LINE;     出列號，並將其加1。）
290 LINE = LINE + 1
300 PRINT "POKE ";PLACE;" , "; PEEK
      (PLACE);" :";         （存入一位元組，處理下一個
310 NEXT PLACE              位址。）
320 PRINT                   （建立新的一列以便加入DOS
                            指令）
330 PRINT D$;"CLOSE"        （然後關閉檔案）
340 END
```

欲利用此程式，請將**SAMPLE PROGRAMS**磁碟片放入磁碟機，並打入

```
RUN POKER
```

當POKER詢問時，輸入您的指令檔之檔名（列 130），輸入記憶體位址（列 140和 150），並輸入 BASIC 的起始列號（列 170）。如果您取其名爲FILLER，可用如下指令執行：

```
EXEC FILLER
```

FILLER會將這些 POKE 指述一一加入記憶體的 BASIC 程式中。

5.2 EXEC指令

EXEC指令告訴 DOS 由指令檔中讀取指令，而不再從鍵盤接受指令。您可以從指令檔的任一列指令開始執行。

EXEC指令的格式如下：

EXEC fn [,Rn] [,Sn] [,Dn] [,Vn]

fn　　指明指令檔之檔名；它係一文字檔，可包含 DOS 指令、資料和 BASIC 指述。

[,Rn]　　其中 n 是由 0 至 32767的任一數目，指出要越過指令檔幾列指述後才開始執行。如果 Rn 省略，則 DOS 從第 1 列開始執行指令檔。DOS從指令檔的開頭計算資料欄（field），所以 Rn 都是相對於 0 來計算的。如果指定的 R 值超出檔案末尾，則執行的結果會印出 END OF DATA 之錯誤訊息。

[,Sn]　　其中 n 是由 1 至 7 的任一數目，指定所使用的磁碟機其磁碟控制卡所在的擴充槽。如果省略此項，DOS 採用預設槽號（default slot number）。

[,Dn]　　其中 n 是 1 或 2 ，指定存此指令檔的磁碟所在的磁碟機

。如果省略此項，DOS採用預設磁碟機代號（default drive number）。

[,Vn] 其中n是由0至254之任一數目，指定所用的磁碟機之卷號（volume）。如果您指定此項爲0、或只打入V、或省略此項，DOS就不管卷號。某一時刻內，只能執行一個指令檔。

警告：必須注意，如果指令檔存於某一磁碟機的磁碟上，且指令檔欲執行（run）的某程式存於另一部磁碟機的磁碟片上，則存有EXEC檔案的磁碟片必須放在預設磁碟機（default drive），如此做的原因是程式各部份執行完之後，才能跳回EXEC檔案。

當指令檔完成內含的全部指令後，自動關閉此檔並停止執行。如果指令檔中又使用EXEC指令叫用其他指令檔，則原有的指令檔關閉，新的指令檔開啓並執行之。第二個指令檔取代第一個指令檔的記憶位址。任何時候，存放於記憶體的目前指令檔，不受NEW指述、或FP、INT、CLOSE等指令的影響。

在指令檔中使用EXEC指令，不會消除記憶體中已有的程式。

您可以停止正在執行BASIC程式的指令檔，按CONTROL-C即可；指令檔的其餘指令自動放棄執行。如果要停止正在執行DOS指令的指令檔，則只有關掉電腦。

警告：當指令檔正在執行程式時，程式中的INPUT指述係由指令檔讀取下一欄資料，而不是從鍵盤。如果輸入的是一個立即執行模式（immediate execution）的DOS指令，則此指令會在程式繼續之前先行執行。

5.3 MAXFILES指令

在啓動系統後，DOS 允許您最多同時開啓 3 個檔案。但是，如果您要建立一個大程式，可能需要開啓更多的檔案。或者，您要充份利用記憶體中每一個位元組（byte），這時候您必須減少同時開啓的檔案數。

MAXFILES 指令讓您增加“現用檔案”（active file）至16個，或者減至只剩一個。利用本指令所指定的最大值一直有效，直到下次再用 MAXFILES 指令重設或重新啓動 DOS 爲止。

MAXFILES 指令的語法如下：

```
MAXFILES n
```

其中 n 是 1 至 16 個，指定任一時刻最多之檔案保留區（file buffer）的個數。如果您指定的值超出此範圍，畫面出現如下的錯誤訊息 RANGE ERROR

5.3.1 保留區的資訊

對於每個您開啓的檔案，DOS 都會騰出 595 位元組的記憶空間，稱做**檔案保留區**（file buffer）。DOS 使用保留區中的 256 個位元組來存放資料（data），並用 256 個位元組來記錄磁軌／磁區使用表（track / sector list），其餘 83 個位元組用來記載雜項，如檔案於磁碟中的位址。

當您的程式從磁碟讀取資訊時，DOS 讀進 256 個位元組，並將其放入保留區中的資料部份（data section）。每當您的程式要求這

註：有關磁軌／磁區使用表，請參考附錄 B。

256個位元組的某一部份時，DOS將所要求的資料交給您的程式。

當您的程式將資料送給磁碟時，DOS先將這些字元放入保留區中的資料部份，直到累積至256個位元組，DOS才將這些資訊一次寫入磁碟。

順便一提：DOS一開始將保留區的資料部份256位元組全部塡0，然後才一一被您寫入的資料所取代。當DOS將保留區的資料寫入磁碟時，係將整個256位元組轉入磁碟。如果您寫入的資料少於256位元組，則DOS只傳送您所寫入的資料，其餘的部份仍留在保留區上。

DOS在啓動之後提供三塊檔案保留區，總共預留1785位元組的記憶空間。當您用MAXFILES指令來擴充檔案個數時，DOS對於每一新檔案增加595位元組的記憶空間做爲保留區。當您用MAXFILES減少“現用檔案”個數時，DOS對於每個不用的保留區釋放595位元組回到記憶空間。

您可以在載入及執行程式之前，以立即執行（immediate execution）的方式下達MAXFILES。

警告：當您從鍵盤下達MAXFILES時，DOS移動記憶體中的HI-MEM：位址值，但是不移動任何Integer BASIC 程式或Applesoft字串。
變更保留區數量，會消除Integer BASIC 程式。增加保留區的數量會影響Applesoft 字串儲存的區域。

順便一提：您也可以在Applesoft 程式中使用MAXFILES指令。

註：1位元組（byte）＝1字元（character）。

註：檔案保留區（file buffer）＝595位元組的記憶空間。

註：HIMEM:＝Applesoft程式可用的最高記憶位址；請參考第6章。

在程式的第一個指述就使用MAXFILES指令，因它會變更字串變數的記憶指標(memory pointer)，相當於重新設定(reset)程式的字串變數。

除PR#、IN#和MAXFILES之外的所有DOS指令，都會用到檔案保留區。如果您設定MAXFILES是1，並且開啓一檔案，則您已用完極限——“一塊保留區”。如果接著您又用到PR#、IN#和MAXFILES之外的DOS指令，畫面將出現 NO BUFFERS AVAILABLE，表示沒有保留區可用。

5.3.2 範 例

要在Applesoft程式中變更MAXFILES，可在程式的第一個指述，在宣告任何字串變數之前，使用MAXFILES指令。例如，

```
10 PRINT CHR$(4); "MAXFILES 5"
```

5.4 CHAIN

CHAIN用來將一程式和另一程式連接在一起。當兩份程式都是以Integer BASIC寫成，則可利用“CHAIN指令”連結。當兩份程式都是以Applesoft寫成，則可利用SYSTEM MASTER磁碟片的“CHAIN程式”加以鏈結(chain)。

5.4.1 CHAIN指令——Integer BASIC

當一份Integer BASIC程式過大而無法容入記憶體時，CHAIN指令可將程式各部份分別載入，依次執行各部份。當第二

部份進入時，DOS不會清除第一部份所用的變數或關閉其所用的檔案。

指令格式如下：

CHAIN fn [,Sn] [,Dn] [,Vn]

fn	指定下一個要執行的 Integer BASIC 模組（module）所在的檔案。
[,Sn]	其中n是1至7的任一數字，指定要使用的磁碟機其磁碟機控制卡所插的擴充槽號。如果此項省略，DOS採用預設槽號。
[,Dn]	其中n是1或2，指定存放檔案 fn 的磁碟片其所在磁碟機。如果此項省略，DOS採用預設磁碟機號碼。
[,Vn]	其中n是0至254之任一數目，指定使用的磁碟機之卷號。如果您指定爲0或只打V或省略此項，DOS使用預設卷號。

當您由一程式鏈結到另一程式，第一個程式內容從記憶體中移走，執行程序從下一個程式編號最小的指述開始執行。假若希望再使用第一個程式，可再使用CHAIN 指令一次。

警告：如果有一陣列（array）同時在兩個鏈結程式中，必須在含此陣列的程式中定義其維數（dimension）。

如果兩個鏈結程式都使用同一個定義函數（DEF function），必須確保它在兩者中皆有定義。

在下例中，程式PART 1使用CHAIN 指令來連接第2份程式PART 2。

記住：必須確定您是在執行 Integer BASIC，如果沒有看到提示記號>，要先下達INT指令

1. 打入此程式，在列 5 0 的第一個雙引號（quotation mark,"）之後須打入 CONTROL - D 字元。

```
10 REM PART1
12 DIM I$(30)                （Integer BASIC中，字串
                             變數若要存一個以上字元，則
                             必須宣告其大小。）
30 I$="THE STRING I$ IS PRESERVED."
                             （設定字串的值）
40 PRINT "PART1: I$ HAS BEEN SET."
                             （宣告完成）
50 PRINT "CHAIN PART2 "
                             （鏈結 PART 2）
60 END
```

要保存此程式，請打入

```
SAVE PART1
```

2. 現在打入**PART 2**程式，並將其保存起來。

```
15 REM PART2
45 PRINT "PART2: ;I$    （顯示PART1 所設定的字串）
60 END
```

3. 如果您要看看輸出是什麼，請打入

```
RUN PART1
```

如果**CHAIN** 指令執行無誤，則當**PART 2**印出變數 I$ 時，其值必然與**PART 1**設定的值一致，畫面將會出現 `PART2; THE STRING I$ IS PRESERVED.` 。如果**CHAIN** 並沒有發生作用，則畫面只出現 `PART2:` 。

5.4.2　CHAIN 程式——Applesoft

本例子利用 SYSTEM MASTER 磁碟中機器碼型態（ B 型態）的 CHAIN 程式，來說明如何鏈結 Applesoft 程式。PART1 程式設定變數 I$ 的值，然後使用 CHAIN 程式與 PART2 鏈結起來。

記住：必須確定存有 PART 1 和 PART2 的磁碟，同時也有 CHAIN 程式。如果沒有，可用 FILEM 程式由 SYSTEM MASTER 磁碟片抄過來。

1. 打入 PART 1 並保存起來。

警告：當您在輸入列 60 時，不要在數目字 520 之後加入空白字元（ space ）、逗號（，）、冒號（：）或分號（；），520 是 CHAIN 副程式載入的記憶位址。

```
10 REM PART1
20 D$=CHR$(4)                    （D$設成CONTROL-D）
30 I$="THE STRING I$ IS PRESERVED."
                                 （設定字串值並加以顯示
                                 出來）
40 PRINT "PART1: I$ HAS BEEN SET."
50 PRINT D$;"BLOAD CHAIN, A520"
                                 （載入記憶位址）
60 CALL 520"PART2"               （鏈結PART 2）
```

2. 現在輸入 PART2 並保存起來。

```
15 REM PART2
45 PRINT "PART2: ";I$
60 END
```

PART 1 的列 50 中使用 CHAIN 程式，使 PART 2 能顯示 I$ 的內容（列 45）。

3. 要執行此示範例子，可打入

```
RUN PART1
```

如果CHAIN程式工作正常，則當PART2印出I$的值時，其值應當與PART1所設定的值一致，印出的結果是PART2; THE STRING I$ IS PRESERVED。如果CHAIN沒有發生作用，則只印出 PART2:.。

5.5 MASTER程式

每當您啓始(initialize)一磁碟片時，都會自動抄一份DOS的副本存入此新的磁碟片。磁碟啓動(start up)時，DOS載入記憶體的位址，與啓始此磁碟片的系統相同。如果您在不同記憶體容量的電腦啓動此磁碟片，DOS無法工作。因此，如果您使用數台Apple II電腦，並且各具有容量不同的記憶體，主磁碟片(master disk)就非常有用。

主磁碟中含有可重定址(relocatable)的DOS機器碼(image)，可使用於16K以上的任何Apple II電腦。當啓動主磁碟時，DOS載入的位址，係根據使用此磁碟片的系統之大小來決定。

MASTER是一個Applesoft程式，用以執行MASTER CREATE，後者是SYSTEM MASTER磁碟片上的"機器碼程式"(binary program)，將已啓始化(initialized)、可寫入資料的磁碟片轉換成主磁碟片，轉換此磁碟片不會損害到此磁碟片上其他程式。

MASTER會請求輸入一程式名稱，爾後每次啓動此磁碟時，DOS自動執行此程式。您可以採用傳統的歡迎詞程式(greeting program)或其他應用程式。

註：如果您並不需要在記憶體容量較小的電腦中使用您的磁碟片，可以略去本節的內容。

順便一提： 早期的手册已指明要使用“機器碼指令”(binary command)來執行MASTER CREATE。現在您可以直接打入如下的指令來執行：

```
BRUN MASTER CRE/
```

警告： 在使用MASTER 之後，在做其他工作之前，經常是需要重新啓動DOS。

5.5.1　範　例

1. 寫一個歡迎詞程式，此程式可表明“主磁碟狀態”(master status)。例如：

```
10 PRINT "MASTER DISK 32K"
20 END
```

2. 將此程式保存到您所要轉換的磁碟，可打入

```
SAVE HELLO
```

當使用中(IN USE)的指示燈暗了之後，將磁碟片從磁碟機中抽出。

3. 將 SYSTEM MASTER 磁碟片插入1號磁碟機，並輸入 RUN MASTER 。

畫面出現 EXECUTING MASTER CREATE 的訊息，然後會出現MASTER CREATE的標題畫面：

```
DOS 3.3     MASTER - CREATE UTILITY
COPYRIGHT APPLE COMPUTER,INC. 1980
      ALL RIGHTS RESERVED.

     (NOW LOADING DOS IMAGE)
```

4. 畫面出現 PLEASE INPUT THE "GREETING" PROGRAM'S FILE NAME: 時，必須回答此問題，請輸入您的歡迎詞程式之名稱，然後畫面如下：

```
REMEMBER THAT "MASTER" DOES NOT CREATE
THE "GREETING" PROGRAM, OR PLACE IT IN
THE DISK DIRECTORY.

  THIS IS THE FILE NAME THAT WILL BE
PLACED WITHIN THE IMAGE:


       name

  PLACE THE DISKETTE TO BE MASTERED IN
THE DISK DRIVE.

  PRESS [RETURN] WHEN READY.

NOTE: IF YOU WANT A DIFFERENT FILE NAME,
PRESS [ESC]
```

如果您希望每次啓動磁碟所執行的程式不是畫面顯示的" name " 則按下 ESC 鍵，並輸入您的歡迎詞程式之名稱。

5. 將 **SYSTEM MASTER** 磁碟片從第一部磁碟機抽出，並插入您希望轉換成主磁碟的已啓始化磁碟片，然後按下 RETURN 鍵。**MASTER** 開始從事轉換的工作，而且完成後會通知您。

6. 當轉換工作完成後，磁碟機上的指示燈熄滅，這時可按下 ESC 鍵，抽出磁碟片並貼上標籤，表示這片是主磁碟。

注意：千萬記住，使用**MASTER** 程式之後，必須重新啓動**DOS**。

5.6　製作轉鑰式磁碟

轉鑰式磁碟（turnkey disk）在啓動 DOS 時，會自動執行一程式。本節告訴您如何產生三種轉鑰式磁碟：簡單式啓動磁碟、主啓動磁碟（master startup disk）、以及可以將不同的 BASIC 載入所有不同記憶體大小的Apple II系統磁碟。

警告：轉鑰式磁碟無法自動從 DOS 回到 BASIC，除非轉鑰程式提供“跳出指令”（exit）。

欲製作轉鑰式磁碟，必須將您的轉鑰程式（turnkey program）載入記憶體，並將其保存到已啓始化的磁碟。當您下達 SAVE 令時，採用磁碟的歡迎詞程式之名稱。（譯註：英文原名 turnkey 具有“扭開鑰匙即有作用”這一層含意，故直譯爲“轉鑰式”。實際上啓動電腦的動作，是有如放入一把鑰匙，以發動電腦這個機器。）

5.6.1　建立轉鑰式磁碟

本例用以說明如何建立轉鑰式磁碟，使得每次啓動磁碟時會自動執行RENUMBER程式。

1. 啓始一空白磁碟，使用HELLO做爲歡迎詞程式的名稱。
2. 將 SYSTEM MASTER 插入第一部磁碟機，將RENUMBER程式載入記憶體，請打入

```
LOAD RENUMBER
```

3. 將 SYSTEM MASTER 磁碟片從磁碟機中抽出，將已啓

始的磁碟片插入您的磁碟機。將RENUMBER程式放入此磁碟片，請輸入

```
SAVE HELLO
```

現在RENUMBER程式已取名爲HELLO。每當您啓動此磁碟片，即自動啓動HELLO，此程式即是RENUMBER。

5.6.2 建立主轉錀式系統

利用MASTER 程式將已啓始化的磁碟轉成主磁碟。當MASTER要求您輸入歡迎詞程式之名稱，請輸入該程式之檔名。

5.6.3 萬用磁碟

如果您要製作一轉錀式磁碟，使其可在所有Apple II電腦上執行，您將需要一個DOS，可以在任何大小的記憶體中使用，可以帶入任何您需要的 BASIC 系統。首先必須瞭解，各種類型的Apple II電腦如何與記憶體和 BASIC 互相作用。

當您將 DOS 磁碟插入第一部磁碟機並打開電源之後，您的電腦會執行此磁碟中的**歡迎詞程式**，假設此程式所用的 BASIC 程式在唯讀記憶體 (ROM) 中，亦即是

- 標準的Apple II執行 Integer BASIC 的歡迎詞程式。
- Apple II Plus 和Apple IIe 執行 Applesoft 的歡迎詞程式。
- 具有Applesoft Card 的Apple II和具有 Integer

註: MASTER 程式，請參考5.5.1節。

BASIC Card 的 Apple II Plus 可各自執行界面卡上 BASIC 所寫的歡迎詞程式。

當歡迎詞程式係用不同的 BASIC 來撰寫時，如果此 BASIC 語言不在記憶體中，畫面出現 LANGUAGE NOT AVAILABLE 以及常駐 BASIC 的提示字元（] 或 > 符號）。

如果您有 Apple Language System，則將不同的 BASIC 載入語言界面卡（Language Card）之後，即可執行不同 BASIC 的程式。

順便一提：裝設語言界面卡，會增加 Apple II 16K 位元組的記憶體。在裝設 Language Card 之前，您的電腦必須至少 48K。

載入不同 BASIC 語言的最簡單方法是，利用 SYSTEM MASTER 磁碟片啓動 DOS。SYSTEM MASTER 中的 HELLO 程式，利用 LOADER.OBJ0 來檢查那一種 BASIC 語言係常駐（resident）於記憶體。然後，再將另一個 BASIC 語言由磁碟中載入。

5.6.3.1 範例

本例在說明如何製作可在任何 Apple II 系統啓動的磁碟。

記住：如果您使用一部以上的磁碟機，則必須指定磁碟機選擇項（drive option）[,Dn]。

1. 將 HELLO 程式由 SYSTEM MASTER 磁碟片載入記憶體。當您利用 INIT 指令啓始一空白磁碟時，可採用此

註：可利用 SYSTEM MASTER 來載入不同的 BASIC 語言系統。

HELLO程式做爲歡迎詞程式。INIT 結束之後，再插入 SYSTEM MASTER 磁碟片。

2. 利用 FILEM 來複製以下語言檔與程式，將 BASIC 語言系統從 SYSTEM MASTER 載入您的新磁碟：

```
APPLESOFT
INTBASIC
FPBASIC
LOADER.OBJ0
```

當您啓動此磁碟時，DOS將會載入不同的 BASIC 。然後，將其他程式放入此磁碟片。

3. 利用 MASTER 程式，將已啓始的磁碟轉換成主磁碟。請打入

```
RUN MASTER
```

當 MASTER 的提示出現時，將新啓始的磁碟插入磁碟機。當 MASTER 問及歡迎詞程式之名稱時，請打入

```
HELLO
```

4. 修改HELLO程式，以便執行您的轉鑰程式，在本例中此程式稱爲TURNKEY。

欲修改HELLO，先解鎖（unlock）此檔，載入（load）此程式，並列印如下：

```
10 TEXT : HOME
20 D$ = CHR$ (4) . REM CTRL-D
30 VTAB 2:A$ = "APPLE II": GOSUB 1000
40 VTAB 4:A$ = "DOS VERSION 3.3 SYSTEM
     MASTER": GOSUB 1000
50 VTAB 7:A$ = "JANUARY 1, 1983": GOSUB 1000
60 PRINT D$;"BLOAD LOADER.OBJ0"
```

```
70 CALL 4096
80 VTAB 10: CALL - 958:A$ = "COPYRIGHT APPLE
     COMPUTER,INC. 1980,1982 ": GOSUB 1000
90 C = PEEK ( - 1101): IF C=6 THEN PRINT :
     INVERSE :A$ = "BE SURE CAPS LOCK IS
     DOWN": GOSUB 1000: NORMAL
100 PRINT CHR$(4);"FP"
1000 REM CENTER STRING A$
1010 B = INT (20 - ( LEN (A$) / 2)): IF B =<0
       THEN B=1
1020 HTAB B: PRINT A$: RETURN
```

5. 將列100 改成如下：

```
100 PRINT CHR$(4);"RUN TURNKEY"
```

然後以**HELLO**爲檔名保存（save）起來，並鎖住此檔。

6. 解鎖（unlock）**APPLESOFT** 檔，載入並列印如下：

```
10 TEXT : CALL -936: DIM A$(40)
20 D$="":                          REM CTRL-D
30 VTAB 2:A$="APPLE II": GOSUB 1000
40 VTAB 4:A$="DOS VERSION 3.3 SAMPLE
     PROGRAMS": GOSUB 1000
50 VTAB 7:A$="JANUARY 1, 1983": GOSUB 1000
60 PRINT D$;"BLOAD LOADER.OBJ0"
70 CALL 4096
80 VTAB 10: CALL -958:A$="COPYRIGHT APPLE
     COMPUTER, INC. 1980, 1982": GOSUB 1000
90 C=PEEK (-1101): IF C<>6 THEN GOTO 100:
     PRINT
95 POKE 50,127:A$ ="BE SURE CAPS LOCK IS
     DOWN": GOSUB 1000: POKE 50,255
100 PRINT D$;"INT"
1000 REM CENTER STRING A$
1010 B = 20 - ( LEN (A$) / 2): IF B = 0 THEN
       B=1
1020 TAB B: PRINT A$: RETURN
```

修改列 100，

```
100 PRINT D$;"RUN TURNKEY"
```

以 APPLESOFT 爲檔名，保存此檔，並設定鎖住狀態。更改HELLO和 APPLESOFT，使它可執行您的轉鑰系統，而非原先執行中的 BASIC 系統。

7. 將您的轉鑰程式取名爲TURNKEY。以下的例子是用 Applesoft 所寫的TURNKEY程式：

```
100 TEXT : HOME
200 VTAB 3: PRINT "THIS IS MY TURNKEY
      PROGRAM "
300 VTAB 5: FOR A = 1 TO 10: PRINT A: NEXT A
400 PRINT : PRINT "THIS IS THE END OF MY
      TURNKEY PROGRAM"
500 END
```

8. 確定您的磁碟有HELLO、APPLESOFT、INTBASIC、FPBASIC、LOADER. OBJ0和TURNKEY檔案，檢驗您的轉鑰式磁碟，將它放入第一部磁碟機，關掉電源並重開。磁碟重新啓動DOS，若有需要則載入 Integer BASIC 或 Applesoft ，然後自動執行轉鑰程式 TURNKEY。

5.7 摘 要

5.7.1 指 令

EXEC fn [, Rn] [, Sn] [, Dn] [, Vn]

指示 DOS 從指令檔 (command file)讀取指令，而不是由鍵盤。fn 引數指定一指令檔，該檔係由 BASIC 程式所建立。指令檔可包含 DOS 指令、資料和 BASIC 指述。

MAXFILES n

其中 n 是由 1 至 16 的任一整數，指定在任一瞬間可同時開啓

使用的檔案個數。

CHAIN fn [, Sn] [, Dn] [, Vn]

從磁碟中載入並執行一 Integer **BASIC** 程式，而不至於清除任何變數或陣列在各記憶位址的值，也不會關閉任何已開啓的檔案。

5.7.2　程　式

MASTER

將已啓始的磁碟 (initialized disk)轉換成主磁碟 (master disk)，使其可在任何不同記憶體大小的**Apple II**系統使用。

6 有效使用記憶體

6.1 二進制機器碼檔案 139
6.1.1 二進制位址 140
6.1.2 An 與 Ln：記憶體位址引數 141
6.2 二進制機器碼指令 141
6.2.1 BRUN指令 143
6.2.2 BLOAD 指令 144
6.2.3 BSAVE 指令 145
6.2.3.1 範例 146
6.3 DOS 與監督程式 147
6.3.1 監督程式的輸入與輸出暫存器 148
6.3.2 DOS 的輸入與輸出暫存器 150
6.4 記憶體的使用與進入點 152
6.4.1 記憶體中的 PR# 和 IN# 152
6.4.2 記憶體配置圖 154
6.4.2.1 HIMEM 154
6.4.2.2 HIMEM的值 157
6.4.2.3 高解析度繪圖 158
6.4.3 DOS的進入點 158
6.4.3.1 重新接通 DOS 的副程式 159
6.4.3.2 BLOAD之後的二進制機器碼檔案的記憶體位址 159
6.4.3.3 BLOAD 之後的DOS的記憶體位址 160
6.5 二進制機器碼檔案指令摘要 161

第6章

有效使用記憶體

本章描述如何使用二進制機器碼程式（ binary program），它們在記憶體中如何與監督程式（ Monitor program ）相互作用。內容包括：

- 載入（ load ）、執行（ run ）、保存（ save ）二進制機器碼程式。
- 利用二進制機器碼程式來讀寫字元。
- 控制 DOS 和監督程式的輸入與輸出暫存器（ register）。

以組合語言（ assembly language ）和機器語言（ machine language ）撰寫程式的程式師必須使用二進制機器碼指令（ binary command ）。有關這些語言的使用，請參考 Apple II Reference Manual 和 6502 Assembler／DOS Tool Kit （正式名稱是 Apple 6502 Assembler／Editor Manual ）有更詳細的使用說明。

6.1 二進制機器碼檔案

二進制機器碼檔案 （ binary file ）的資訊，在 Apple II 記憶

體和磁碟上都存放在連續的位址上。資訊不是以文字的形式來表示，而是一連串由 $00 至 $FF 的十六進位數字（十進位 0 至 255）所組成。這些十六進位數字可以表示數字、字母、機器語言指令（machine-language command）或高解析度圖形（high-resolution graphics）。（譯註：雖然 binary file 其性質除了“機器語言”外尚可有其他型式的“資料”。本中文書暫以“二進制機器碼檔案”代表 binary file。）

您可將數值存入二進制機器碼檔案，最大可存到 Apple II 記憶體的上限。二進制機器碼檔案型態是 B。（譯註：另有一標示“R”型態的“可重定位（relocatable）的機器語言程式檔，通常是指經過“Assembler”（組合程式）產生的機器碼，惟此程式檔可以擺放在記憶體的任何位置。B 型態的檔案，若屬於機器語言的程式檔，其載入（BLOAD 指令，見後說明）位址是與儲存的位址（BSAVE）息息相關，固定在某個起始位置，這是兩者的一個差異點。同時，B 型態的檔案，其內容可以是高解析度的“圖像資料”。）

6.1.1 二進制位址

您無須瞭解Apple II記憶體的架構，就可以執行機器語言程式。但是，您必須知道記憶體位址，才能直接使用記憶體中的二進制資訊。您的電腦記憶體是一連串連續的記憶位置或位元組（byte），每一個對應一個位址（address）。第一個記憶體位置的位址是 0（以十六進位表示爲 $0000），第二個記憶體位址是 1（$0001），餘此類推。如果您的電腦具有 48K位元組的記憶體，則最後一個記憶位址是 49151（$BFFF）；如果是 64 K，則最後一個隨意存取記憶體（RAM）的位址是 65535（$FFFF）。

註：二進制（binary）是利用數字 0 和 1，以 2 爲底的數目表示法。

註：十六進位數目（hexadecimal number）係利用數字 0 至 9 與 A 至 F，以 16 爲底來表示數目。

6.1.2　An與Ln：記憶體位址引數

要使用二進制機器碼指令（ binary command），您可以利用下列的引數來指定記憶體位址：

An　指定記憶體中的起始位址（ starting address ）。

Ln　指定記憶體位址（位元組）的個數。

將小寫 n 以十進位或十六進位數目取代。例如，要保存記憶體位址由 8192 至 16383 （ $2000 至 $3FFF ）的高解析度圖形畫面，將其存入檔案，必須指定資料的起始位址如

,A8192　或　,A$2000

並指明您要保存的位元組之總數，如

,L8192　或　,L$2000

記住：十六進位數目必須以 $ 起頭。

6.2　二進制機器碼指令

DOS 指令BRUN、BLOAD 和BSAVE 用來處理電腦記憶體中任何部分的二進制資訊。這些指令的 B 字母表示「二進制」（ binary ）。

這一類的指令可以一個位元組接一個位元組，在記憶體和檔案之間傳送資訊。最常用的兩種指令用法是執行機器語言程式和將“二進制映像”（ binary image ）存入記憶體以便顯示高解析圖像於螢幕上。

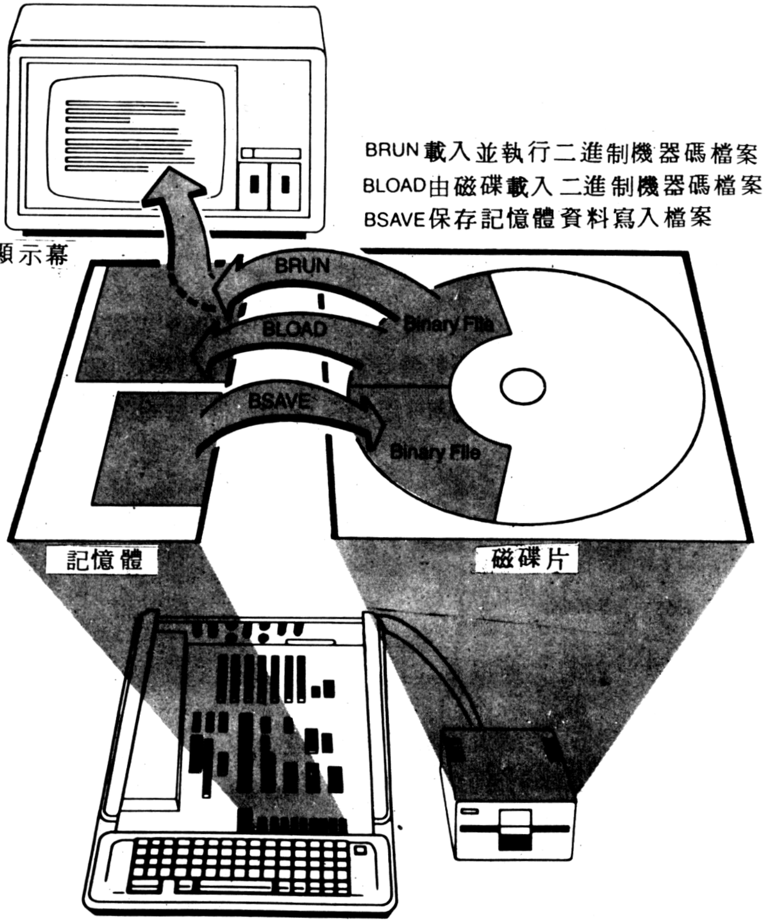


圖6－1 BRUN、BLOAD和BSAVE

6.2.1　BRUN指令

BRUN指令執行二進制機器語言程式檔（檔案型態是B）。當DOS看到此指令，立刻將指定的程式傳送到記憶體，並加以執行。DOS將程式放入記憶體，其起始位址（starting address）係依據程式保存時所定的位址（BSAVE），除非您指定其他的位址。指令的語法：

BRUN fn [, An] [, Sn] [, Dn] [, Vn]

fn　　指定的二進制機器碼檔案。如果您只使用此引數，DOS將fn檔案的整個內容放入記憶體，其起始位址係根據BSAVE指令所指定的位址。

[,An]　其中n是由0至65535（十進位）或$0至$FFFF（十六進位）之間任一記憶體位址，指定DOS傳送程式至記憶體，第1個位元組的位址。

[,Sn]　其中n是由1至7的任一數字，指定所使用的磁碟機之控制卡所在的擴充槽。如果省略此項，DOS採用預設槽號。

[,Dn]　其中n是1或2，指定存有檔案fn的磁碟片所在的磁碟機，如果省略此項，DOS採用預設磁碟機代號。

[,Vn]　其中n是0至254的任一數目，指明所使用的磁碟片之卷號。如果您指定此項爲0，或只打入V，或省略此項，DOS不管卷號。

要執行二進制機器碼程式FID（它是可同時拷貝數個檔案的公用程式），請打入指令

```
BRUN FID
```

順便一提： DOS 無法分辨二進制程式檔（ binary program file ）和二進制資料檔（ binary data file ）。因此，最好在替二進制檔案取名時能提示其內容性質。例如，檔名的末尾是 .PIC 之檔案，表示此檔存放圖像資料（ picture data ）。

如果意外執行（BRUN）一資料檔，可能會更動 DOS（存放於記憶體內）的某些部分。在此種情況下，須重新啓動 DOS 。

6.2.2 BLOAD指令

利用BLOAD指令，可以將一個圖像（ picture ）由檔案移到圖形畫面上，或者將任何型態的檔案中“二進制映像”（ binary image ）移入記憶體，或者將以機器語言撰寫的程式由磁碟檔案傳送到記憶體。語法如下：

BLOAD fn [, An] [, Sn] [, Dn] [, Vn]

fn 指定一個二進制機器碼檔案（ binary file ）。

[,An] 其中 n 是由 0 至 65535（十進位）或由 $0 至 $FFFF（十六進位）之間任一記憶體位址，指定 DOS 傳送此檔案整個內容，第一個位元組在記憶體的位址。假若省略此項，則 DOS 將檔案傳送至記憶體的起始位址，設定成先前用BSAVE 指令儲存檔案的起始位址。

[,Sn] 其中 n 是由 1 至 7 的任一數字，指定所使用的磁碟機之控制卡所在的擴充槽。如果省略此項，DOS 採用預設槽號。

[,Dn] 其中 n 是 1 或 2 ，指定存有檔案 fn 的磁碟片所在的磁碟機。如果省略此項， DOS 採用預設磁碟機代號。

[,Vn]　其中n是0至254的任一數目，指明所使用的磁碟片之卷號。如果您指定此項爲0，或只打入V，或省略此項，DOS 不管卷號。

與LOAD指令不同，BLOAD不會消除記憶體中的程式或變數，除非這些程式或變數所佔的位址，正是BLOAD放入二進制資料的位址。

警告：一個機器語言程式如果移入記憶體的位置不是它當初保存（BSAVE）時所指令的位置，則很可能無法執行。因此，最好將BSAVE指令所用的位址，記錄在磁碟的標籤上。

6.2.3　BSAVE指令

BSAVE指令可將電腦記憶體中任何部分的二進制資訊原原本本傳送到磁碟檔案，使您能夠將機器語言程式由記憶體搬到檔案，或者將圖像從畫圖螢幕存入檔案，或將記憶體任何部分的資訊搬至磁碟的二進制機器碼檔案。

指令的一般格式爲

BSAVE fn, An, Ln [, Sn] [, Dn] [, Vn]

fn　指定一個二進制機器碼檔案。

,An　其中n是由0至65535（十進位）或$0至$FFFF（十六進位）之間任一記憶體位址，它是準備傳送至磁碟的程式在記憶體的第一個位元組位址。這個"起始位址"

註：BLOAD指令只是將二進制資訊由磁碟檔案，傳送到隨意存取記憶體（RAM）。

引數是必要的，不可省略此項。

,Ln　其中n是0至32767（十進位）或$0至$7FFF（十六進位）之間任一數目，表示要由記憶體中傳送出來的位元組總數，這個“長度”引數必須指明。

[,Sn]　其中n是由1至7的任一數字，指定所使用的磁碟機之控制卡所佔的擴充槽。如果省略此項，DOS 採用預設槽號。

[,Dn]　其中n是1或2，指定存有檔案fn的磁碟片所在的磁碟機。如果省略此項，DOS 採用預設磁碟機代號。

[,Vn]　其中n是0至254的任一數目，指明所使用的磁碟片之卷號。如果您指定此項為0，或只打入V，或省略此項，則 DOS 不管卷號。

記住：在磁碟的標籤上，寫下您保存（BSAVE）的二進制機器碼程式之起始位址。

6.2.3.1 範例

在本例中，您將建立一個二進制機器碼檔案，利用此程式來操作 BSAVE、BRUN 和BLOAD指令。

1. 要進入監督程式（Monitor），打入 CALL -151 ，並按下 RETURN 鍵。
2. 將以下的數值輸入記憶體。注意，全部以十六進位表示。

```
300: 20 80 FE A0 00 B9 16 03 F0 06 20 ED FD
C8 10 F5 20 84 FE 4C D0 03 8D C1 D0 D0 CC C5
A0 AF AF 00
```

註：監督程式存在唯讀記憶體內（ROM），它可控制電腦的各項基本功能。

當您輸入所有數值之後，按下 RETURN 鍵。

3. 要求 DOS 保存這些資料到二進制機器碼檔案，並取名為 AppleII，請輸入

```
BSAVE APPLEII,A$300,L$20
```

An 引數表示保存由位址 $300 起的資料，Ln 引數表示保存長度為 $20 個位元組的資料。

4. 執行此程式，請輸入

```
BRUN APPLEII
```

您將看到 APPLEII 字樣以白底黑字的反白模式（inverse mode）輸出於螢幕上。

5. 也可以透過 BASIC 做相同的事情，請輸入

```
BLOAD APPLEII,A$300
CALL 768
```

6.3 DOS與監督程式

在上例中，您進入監督程式，而後才能輸入二進制機器碼程式。本節列舉一些指令，讓您可用它們進入監督程式和回到 BASIC 語言系統。

要從 BASIC 進入監督程式，請輸入

```
CALL -151
```

您會看到畫面出現星號（asterisk *），這是監督程式的提示字元。DOS 這時仍然處於有效使用狀態（active state），

您可以使用任何“立即執行模式”(immediate execution)的 DOS 指令。例如，CATALOG 指令會在螢幕上列出檔案目錄；PR#指令會執行磁碟中的某個程式。

您可以在監督程式之下，使用下列的指令（ n是槽號），無論 DOS 是否處於有效使用狀態。

n CONTROL - P	（與監督程式中使用PR#n相同一指定輸出裝置）
n CONTROL - K	（與監督程式中使用IN#n相同——指定輸入裝置）
CONTROL - C	（繼續使用 BASIC ，不會重新設定（ reset ） BASIC ）。
CONTROL - B	（進入 BASIC ，並重新設定 BASIC ）
*3D0G	（進入 BASIC ，不重新設定，並再度接通DOS)
3D3G	（進入 BASIC ，重新設定 BASIC ，再度接通 DOS ）
3EAG	（如果 DOS 仍在記憶體中，接通 DOS 後保持在監督程式狀態）

希望選擇輸入和輸出裝置，以及再度接通 DOS （ connect DOS ），您也可以直接利用POKE 指述：將適當的數值直接放入監督程式暫存器（ Monitor register ）。

6.3.1 監督程式的輸入與輸出暫存器

表6－1中， n CONTROL - K 和 n CONTROL - P 是監督程式的指令。

註：當您輸入 CONTROL - K 和 CONTROL - P 這兩個字元時，它們不會顯示在螢幕上。

表6－1：位址 56－57($38－$39)的監督程式輸入暫存器(Monitor Input Register)

暫存器內容的設定	設值結果	隨後輸入的來源
[RESET] 0 [CONTROL] - [K] IN#0	−741 ($FD1B)	鍵盤輸入，由監督程式的“輸入副程式”負責。
n [CONTROL] - [K] IN#n (其中n＞0)	49152+n*256 ($Cn00)	第n槽(slot n) 如果第n槽插有磁碟機控制卡，則啓動 DOS。
DOS啓動(boot)	最高記憶體位址 −8515(−$2143)	DOS

表6－2：位址 54－55($36－$37)的監督程式輸出暫存器(Monitor Output Register)

暫存器內容的設定	設值結果	隨後輸出的對象
[RESET] 0 [CONTROL] - [P] PR #0	−528 ($FDF0)	透過監督程式的“輸出副程式”到顯示螢幕上。
n [CONTROL] - [P] PR#n (其中n＞0)	49152+n*256 ($Cn00)	第n槽 如果第n槽插有磁碟機控制卡，則啓動 DOS。
DOS啓動(boot)	最高記憶體位址 8575($217F)	DOS

6.3.2 DOS的輸入與輸出暫存器

表6－3 DOS 的輸入暫存器（DOS Input Register）

暫存器內容的設定	設值的結果	隨後輸入對象
DOS啓動(boot) [RESET] 3D0G IN#0 PRINT D$;"IN#0" (D$ = [CONTROL] - [D])	−741 $FD1B	鍵盤輸入，由監督程式的"輸入副程式"負責。
IN#n PRINT D$;"IN#n" (其中n＞0)	49152+n*256 ($Cn00)	第n槽 如果第n槽插有磁碟控制卡，則重新啓動DOS。

表6－4 DOS輸出暫存器（DOS Output Register）

暫存器內容的設定	設值結果	隨後輸出的對象
DOS啓動(boot) [RESET] 3DØG PR#Ø PRINT D$;"PR#Ø" (D$= [CONTROL] - [D])	−528 ($FDF0)	透過監督程式的輸出副程式到顯示螢幕。
PR#n PRINT D$;"PR#n" (其中Ø＜n＜8)	49152+n*256 ($Cn00)	第n槽 如果第n槽插有磁碟機控制卡，則重新啓動DOS。

當 DOS 正在操作時，PR #和IN # 不會影響到 DOS 的輸入與輸出暫存器，除非是在指令檔中不以 CONTROL - D 的方式來使用。例如，當指令檔執行某一例如 120 PR#3， ，則監督程式的輸出暫存器有更動，DOS 某些部份功能暫時失效，直到下一次輸入為止。

無論 DOS 的輸入與輸出暫存器所選的是那一個輸出或輸入裝置，DOS 仍舊可自磁碟中接受（輸入）資料，也可以傳送（輸出）資料至磁碟內。

有兩種方法可以選擇輸出入設備及重新接通 DOS ：您可以使用表6－1至表6－4所列的指令（設定方法），或者利用POKE指述直接將適當的值放入 DOS 的暫存器。

警告：DOS 的輸入與輸出暫存器所在的特定記憶體位址，會因不同大小的系統記憶體和不同的 DOS 版本有不同的位址。因此，您必須依照以下特別的兩個步驟來更改這些暫存器位址的內容：

1. 更改監督程式輸入與輸出暫存器，使其值正是 DOS 的輸入與輸出暫存器所要的數值。可以利用POKE指述將此值直接放入監督程式的位址；或者在程式中直接用IN #和PR #，不用 CONTROL - D 的方式。
2. 利用CALL 1002（由 DOS 下指令）或$3EAG （由監督程式下指令）可重新透過監督程式的暫存器接通 DOS 。監督程式的輸入與輸出暫存器其原有的內容會出現在 DOS 的輸入與輸出暫存器。每當您的程式因某種需要暫時切斷 DOS 時，利用此CALL都可重新接通 DOS 。

註：有關上述步驟的例子，可參考第 2 章中尋找隱藏字元的程式。

DOS 無法辨認監督程式指令n CONTROL - K 和 n CONTROL - P，因爲這些指令直接影響監督程式的輸入與輸出暫存器，所以當 DOS 接通後，必須利用PR#和IN#指定輸出和輸入裝置。

6.4 記憶體的使用與進入點

本節描述PR#和IN# 如何使用記憶體，並告訴您在啓動DOS 時記憶區域的使用狀況，DOS 指定給HIMEM的記憶體位址，以及 DOS 和兩種 BASIC 各自使用的記憶區域。

6.4.1 記憶體中的PR# 和IN#

要看看PR#和IN# 如何在記憶體中運作，可檢視 Apple II 平常如何傳送和接收字元。有兩個記憶體位址，分別叫做CSWH和 CSWL，存放“輸出字元的副程式”的進入點位址（entry point）CSWL是**監督程式的輸出環**（monitor output link），其作用是連接監督程式與輸出副程式。

同樣地，有另外兩個記憶體位址，分別叫做KSWH和KSWL，存放“讀取字元的副程式”的進入點位址。KSWH和KSWL 是**監督程式的輸入環**（monitor input link）。

監督程式輸出環通常存放“標準輸出副程式”（standard output routine）COUT1的位址，此副程式將字元送到顯示幕上。監督程式的輸入環通常存放標準輸入副程式（Standard

---

註：CSWH與CSWL＝監督程式輸出環（monitor output link）。

註：KSWH和KSWL＝監督程式輸入環（monitor input link）。

input routine ）KEYIN 的位址，此副程式由鍵盤中讀取字元。

當 DOS 不處於執行中的狀態，而且您在 BASIC 中使用PR#和IN# 時，監督程式還會設成目前所指定的擴充槽內界面卡（card ）的ROM地址（第 n 槽則是 $Cn00 ）。當電腦由鍵盤讀取字元或傳送字元到顯示幕，則呼叫界面卡上的存放於 ROM 的輸出入副程式，來執行傳送的工作。

當 DOS 在執行時，監督程式的輸出入環所存放的位址是DOS的輸出入副程式，而非標準輸出入副程式。 DOS 將監督程式的標準輸入程式之位址保存在 DOS 的輸出入環（ DOS I／O link ），意即 DOS 的輸入環通常存放 KEYIN 和COUT1的位址。

當您利用PR#和IN# 指定槽號時， DOS 將 DOS 環的內容代換成您所指定的擴充槽上卡的 ROM 位址（第 n 槽則是 $Cn00 ）。當您利用PR#和IN# 指定位址時， DOS 將此位址存入DOS環。而後，若 Apple II 要讀寫字元時，監督程式的輸出入環指向適當的 DOS 副程式，這包括兩部分的轉換：

- 第一部分， DOS 將目前的輸出入副程式的位址由 DOS 的輸出入環移到監督程式的輸出入環。然後 DOS 呼叫 COUT1或KEYIN ，由監督程式的輸出入環所選到的裝置負責寫入或讀取資料。
- 第二部分， DOS 重新接通自己，將其輸出入副程式的位址放入監督程式的輸出入環。

假若在程式中下達PR#或IN#指令，並未以 CONTROL - D 前導的方式使用這兩個指令，則立即進入 BASIC 語言系統，隨後 BASIC 更改監督程式的輸出入環之值。因爲監督程式環不再指向 DOS ，表示目前情況下 DOS 沒有接通。由於兩個階段的

註：COUT1將字元送到顯示幕；KEYIN 則由鍵盤讀取字元。

轉換過程， DOS 會先利用舊的輸出入副程式（第 1 部分）來傳送下一個字元。然後，如果輸出入暫存器沒有同時切斷，則會自行接通（第 2 部分）。

假若發生上述情況，須重設 DOS 的輸入和輸出裝置爲鍵盤和顯示幕。如果 DOS 不在記憶體中，重新啓動 DOS 。如果 DOS 仍然在記憶體中，利用 BASIC 指述 CALL 1002 即可。

順便一提：CALL 指述是在 BASIC 之下執行某個特定位址的機器語言副程式，將控制流程轉入副程式。當副程式結束之後，返回執行在CALL指述之後的 BASIC 指述。CALL 不會影響記憶體中的程式內容。

6.4.2 記憶體配置圖

6.4.2.1 HIMEM

HIMEM是 Applesoft 程式內各指述和變數可用的“最高記憶體位址”（ highest memory address ）。Applesoft 自動設定 HIMEM 爲電腦的可用 RAM 中最高的可寫記憶體位址（加 1 ）。DOS 載入之後，自動重設HIMEM 爲較低的值，以保護 DOS 所佔用的區域。根據 DOS 所定的值，HIMEM 位址“以上”的記憶體區域可提供 DOS 、高解析度繪圖或機器語言程式使用。

圖 6 － 2 展示有關啓動 DOS 時記憶體使用狀況。圖 6 － 3 說明有關 DOS 和兩種 BASIC 語言所使用的記憶區。

啓動主磁碟（ master disk）會影響到位址 6912 至 16383 （ $1B00 到 $3FFF ）的內含值。但，如果啓動的磁碟是在 32K 或更大的系統操作啓始化的步驟（即 INIT 指令），則此塊區域不

註：HIMEM=Applesoft 可用的最高記憶體位址（加 1 ）

受影響。DOS 在記憶體中佔用的區域是緊接在最高 RAM 記憶體位址“之下”，此位址是當時啓始該磁碟片的電腦系統內“可使用的最高記憶體位址”。

圖 6－2 啓動 DOS 時的記憶分配

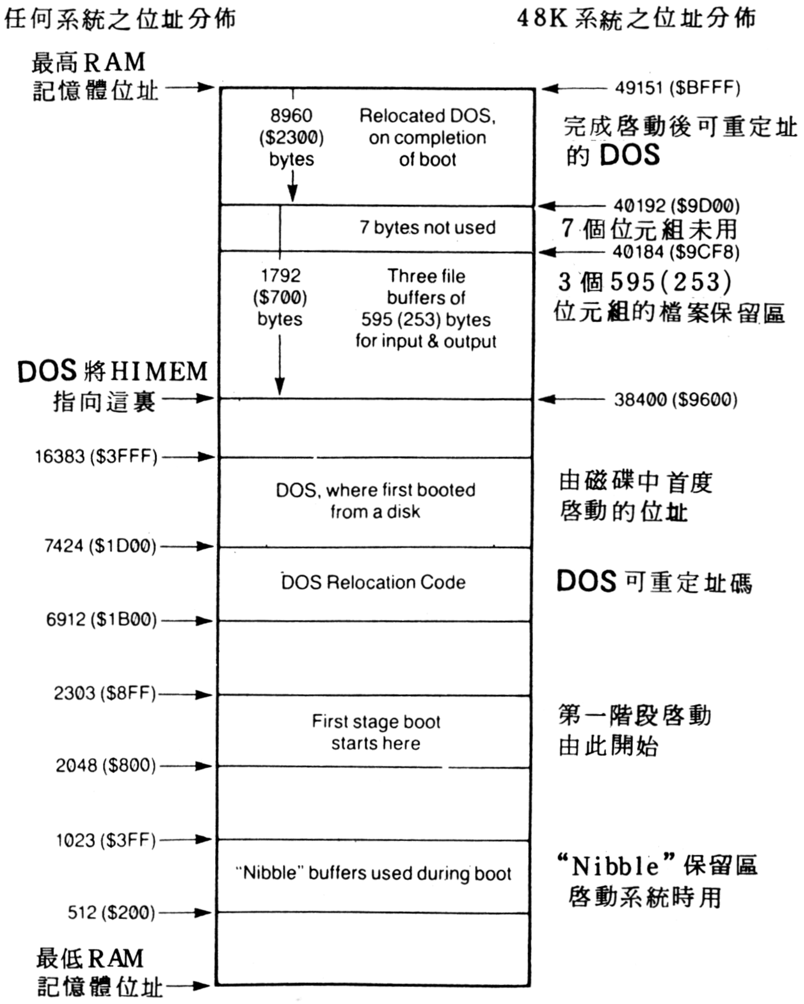
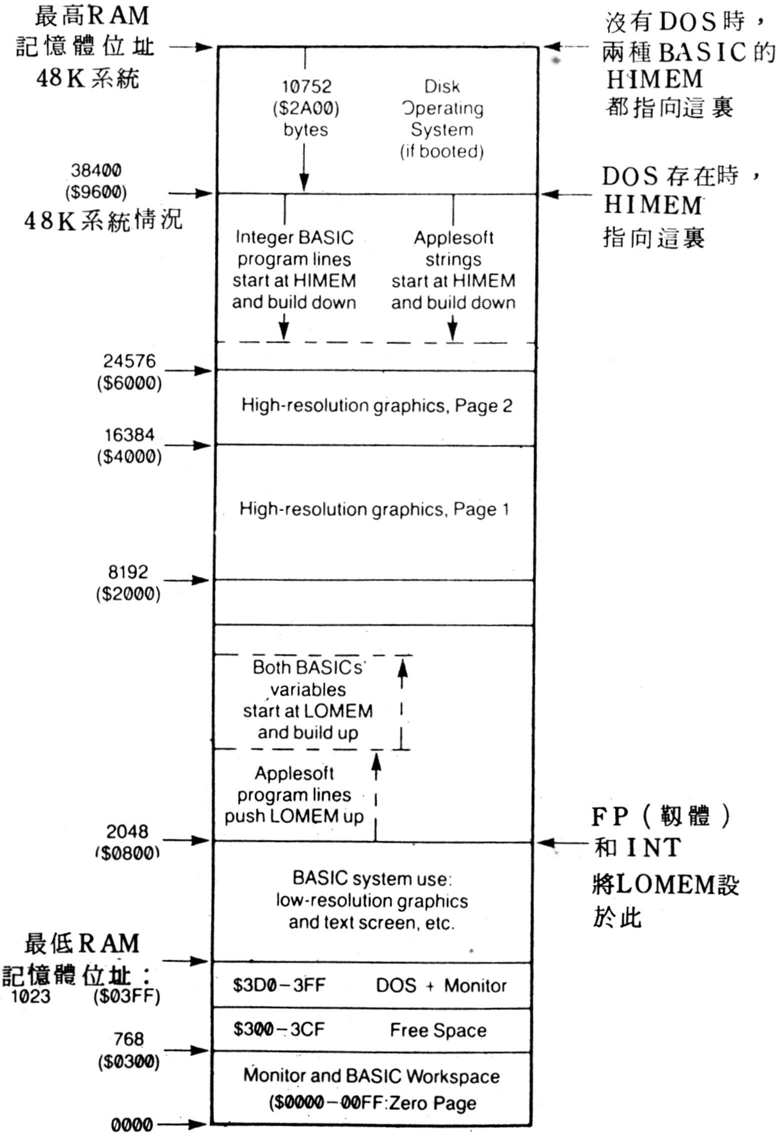


圖 6 － 3 DOS 與兩種 BASIC 所使用的記憶區域



6.4.2.2　HIMEM的值

HIMEM是您的程式可用的最高記憶體位址，其上限由 Apple II電腦在啓動時所設定。啓動 DOS 時設定HIMEM值，係根據您的電腦所擁有的記憶體容量來決定。表 6 － 5 列出不同系統的對應值。當您的系統正在執行 Integer BASIC 時，HIMEM指標值存於（低位元組在前，高位元組在後）位址 76－77（$4C－$4D）。當您的系統執行Applesoft 時，HIMEM指標存在115-116($73－$74)。

增加最高檔案個數（MAXFILES）時，每加入一個檔案保留區，HIMEM 會往下移 595 個位元組。Applesoft Ⅱ BASIC Programming Manual 提供其他Applesoft 程式的指標位址及意義說明，值得有興趣的讀者參考。

6 － 5　HIMEM的值

系統大小	最高RAM位址 十進位	十六進位	HIMEM(DOS 啓動所設) 十進位	十六進位
16K	16383	$3FFF	5632	$1600
20K	20479	$4FFF	9728	$2600
24K	24575	$5FFF	13824	$3600
32K	32767	$7FFF	22016	$5600
36K	36863	$8FFF	26112	$6600
48K (*)	49151	$BFFF	−27136	$9600 (**)

（＊）這些數值適用於 48K 或更大的系統。

（＊＊）十進位數字 －27136 也可以寫成 38400，但是 Integer BASIC 無法接受大於 32767 的數值。

順便一提：在 Integer BASIC 中，記憶體位址大於 32767 者必須以對應的負數來表示。任何正的十進位位址 n 其對應的

負值是（ n－65536 ）

6.4.2.3 高解析度繪圖

Apple II 電腦有兩塊記憶區供低解析度繪圖，以及兩塊區域供高解析度繪圖，這些區域稱爲**分頁**（ page ）。分頁以行列的方式畫在顯示幕上。

使用高解析度繪圖第 1 分頁，會清除位址介於 8192 和 16383 之間的記憶體內容。除非您的系統的記憶體在 32 K以上，否則無法同時使用磁碟與高解析度繪圖。

警告: HIMEM 和LOMEM必須適當設定，以避免和高解析度繪圖第 1 分頁相同。

使用高解析度繪圖第 2 分頁，會清除位址介於 16384和24575 之間的記憶體內容。除非您的系統的記憶體多於 32 K，否則無法同時使用磁碟和高解析度繪圖第 2 分頁。

警告: HIMEM 和LOMEM必須適當設定，以避免和高解析度繪圖第 2 分頁衝突。

6.4.3 DOS的進入點

本節提供下列幾個 DOS 的進入點（ entry point），並說明使用的方法：

- 如果寫入的資料已越過界限（ overwritten），再重新接通 DOS 。
- 由BLOAD指令載入記憶體的二進制機器碼程式，可找出它的起始位址（ starting address ）和長度。

- 由BLOAD指令最近（most recent）載入記憶體的程式，可找出存其起始位址和長度的 DOS 位址。

6.4.3.1　重新接通DOS的副程式

如果您意外地進入監督程式，則第3分頁的 DOS 指標（pointer）和副程式（subroutine）可能會被切斷（disconnect）。第3分頁（Page 3）是保留給 DOS 使用的記憶區域（位址由 $Ø3ØØ 到 $Ø3FF）。

要重新接通（reconnect）DOS，可使用 CALL 或監督程式的G指令，其指定的位址視您的記憶體之大小而定，這些位址如下：

系統大小	十進位位址（CALL）	十六進位位址（G）
48K*	−25153	$9DBF
32K	23999	$5DBF
16K	7615	$1DBF

（*）適用於 48K或更大的系統

利用監督程式指令3DØL，可在螢幕右上端列出此位址值。

6.4.3.2　BLOAD之後的二進制機器碼檔案的記憶體位址

利用BLOAD指令之後，可以找到一個二進制機器碼檔案的起始位址和長度。請打入

```
PRINT PEEK(低位元組位址) + PEEK(低位元組位址+1)*256
```

要知道您的 Apple II 系統對應的低位元組的位址，請利用下表。

系統大小	起始位址（低位元組）十進位	十六進位	程式大小（低位元組）十進位	十六進位
48K*	43634	$AA72	43616	$AA60
32K	27250	$6A72	27232	$6A60
16K	10866	$2A72	10848	$2A60

（*）適用於 48K或更大的系統

6.4.3.3 BLOAD之後的DOS的記憶體位址

由BLOAD指令最近載入記憶體的程式，其起始位址和長度所存放於 DOS 的位址可利用以下的程式找出來：

記憶大小：H和T的值（列 7 和列 8 ）係針對 48 K的 Apple II系統。表 6 － 5 說明您的電腦的正確值。

```
5 REM BLOAD FINDER
7 H = 38400: REM DOS-BOOT HIMEM
8 T = 49152: REM HIGHEST ADDRESS
10 D$ = CHR$ (4): REM CONTROL-D
20  PRINT D$; "BSAVE FOO, A$7777, L$77"
30  PRINT D$; "BLOAD FOO"
40  PRINT D$; "DELETE FOO"
50  FOR I = H +1792 TO T
60  IF PEEK (I) <> 119 OR PEEK (I+1) <> 0 THEN
      NEXT I
70  PRINT "LOCATIONS OF START ADDRESS: ";I;",
      ";I + 1
80  FOR I = H + 1792 TO T
90  IF PEEK (O) <> 119 OR PEEK (I+1) <> 0 THEN
      NEXT I
100 PRINT "LOCATIONS OF LENGTH: ";I;", ";I + 1
```

本程式大概要花 2 分鐘的時間，來尋找您所要的位址。

6.5 二進制機器碼檔案指令摘要

BRUN fn [, An] [, Sn] [, Dn] [, Vn]

將磁碟上的二進制機器碼檔案（檔案型態B）之內容，轉存入記憶體中某個部分，然後執行此程式。如果An（位址）選擇項沒有指定，則程式放入記憶體的起始位址，係根據原先保存（save）時BSAVE指令所指定的位址。

BLOAD fn [, An] [, Sn] [, Dn] [, Vn]

將二進制資訊由磁碟轉入記憶體任何部分。當An（位址）沒有指明時，程式放入記憶體的起始位址，係根據原先保存此檔時BSAVE指令指定的位址。

BSAVE fn, An, Ln, [, Sn] [, Dn] [, Vn]

將二進制資料由指定的記憶區域轉入磁碟的二進制機器碼檔案（檔案型態B）。引數An（位址）和Ln（長度）的組合使您可以傳送記憶體任何部分的資訊。

附　　錄

附錄A：處理13磁區格式的磁碟　167

A.1　轉換13磁區的磁碟：CONVERT 13程式　168

A.1.1　實例　168

A.1.2　通配字元　171

A.2　使用未轉換的13磁區磁碟　172

A.2.1　使用START 13程式　172

A.2.1.1　範例　172

A.2.2　使用BASICS磁碟　173

附錄B：儲存資訊的內部處理過程　175

B.1　磁軌與磁區　175

B.2　檔案磁區的內容　177

B.2.1　各類檔案型態所用的檔案磁區格式　177

B.2.2　磁軌／磁區表　179

B.2.3　磁碟檔案名錄　180

B.2.4　卷目次表　184

B.2.5　磁軌的位元配置表　186

B.3　磁軌與磁區的分配　188

B.4　為磁軌與磁區定址：RWTS副程式　189

B.4.1　範例　190

B.4.2　格式：輸出入區段與設備特性表　192

B.5 第3分頁的DOS 向量 194
B.6 第0分頁的使用情形 196

附錄C：DOS錯誤訊息 199

C.1 DOS錯誤訊息 200
C.2 修正錯誤 201

附錄D：程式 213

D.1 SYSTEM MASTER磁碟内的各種程式 213
D.2 SAMPLE PROGTRAMS磁碟内的各種程式 215

附錄E：DOS操作概念及指令摘要 219

E.1 操作概念 219
E.1.1 冷啓動 219
E.1.2 温啓動（所有Apple Ⅱ機種） 220
E.1.3 啓始化一磁碟片 220
E.1.4 產生一主磁碟 221
E.1.5 解譯DOS 指令 222
E.1.6 磁碟容量 222
E.1.7 檔案型態 223
E.2 指令記號之說明 223
E.2.1 記號意義 223
E.2.2 語法 223
E.2.3 引數 224
E.3 指令摘要 227

E.3.1 存取磁碟內容的非程式化指令 228
E.3.2 控制程式環境的指令 232
E.3.3 循序存取文字檔的程式化指令 235
E.3.4 隨意存取文字檔的程式化指令 238
E.3.5 二進制機器碼檔案的指令 241

附錄 名詞解釋 243

附錄 快速提示表 261

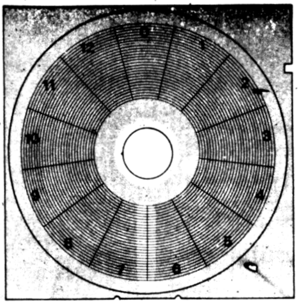
附錄A

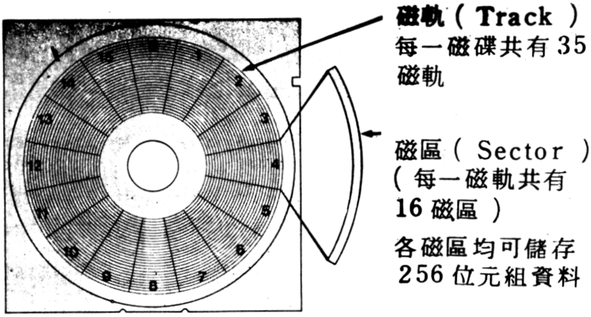
處理13磁區格式的磁碟

本附錄的適用對象是那些手上有遊戲程式或應用程式的磁碟，卻無法在目前的電腦系統正常運轉的使用者。本附錄告訴您如何轉換 13 磁區格式的磁碟成 16 磁區格式的磁碟，並如何在 16 磁區格式的 DOS 之下使用未轉換的 13 磁區磁碟。

早期版本（3.1和3.2）的DOS是將磁碟的表面規劃成每一磁軌（track）含有13個磁區（sector），目前您擁有的DOS 3.3 則規劃成16個磁區（圖A-1），亦即是每一磁碟的可用空間增加約 20%。

假若系統使用了不正確格式的磁碟，DOS會印出適當的錯誤訊息，例如 UNABLE TO READ, UNABLE TO WRITE, or I/O ERROR。


13 磁區格式



16 磁區格式

A.1 轉換13磁區的磁碟：CONVERT13程式

SYSTEM MASTER 磁碟內的CONVERT13程式其作用是將一13磁區的磁碟（包括所含的程式）轉換成16磁區格式的磁碟。CONVERT13 讀取13磁區磁碟的內含資訊，而後寫入另一已啓始化的16磁區磁碟上，其過程中完全不改變任何存於13磁區磁碟的資訊。

附帶一提，較早期版本DOS的MUFFIN程式執行相同於CONVERT13的功能。

A.1.1 實 例

下列的CONVERT13 程式的實例操作中，是假設您的Apple II電腦配備有一部磁碟機，而且磁碟機是接在編號6的擴充槽上的控制界面卡。

1. DOS已存在記憶體內，啓始一空白磁碟片。
2. 將 SYSTEM MASTER 磁碟放入磁碟機，再打入

```
RUN CONVERT13
```

螢幕簡短顯示 EXECUTING MUFFIN 訊息，隨後，出現如圖A-2所示的CONVERT13功能表，

```
***************************************
*  APPLE II DOS 3.2 TO 3.3 CONVERTER  *
*                                     *
*          MUFFIN VERSION D           *
*                                     *
*  COPYRIGHT APPLE COMPUTER,INC. 1979 *
***************************************
```

圖A-2 CONVERT13功能表

打入

1

表示您希望"轉換"（convert）檔案。

3. 當出現SOURCE SLOT?問題後，打入

6

此數目是對應於13磁區磁碟所在的磁碟機，其磁碟機界面控制卡所屬的擴充槽編號。

4. 下一個問題DRIVE?顯示於螢幕時，打入

1

表示磁碟機的代號

5. 當出現DESTINATION SLOT?問題後，打入

6

表示拷貝版磁碟亦共用同一擴充槽。

6. 再顯示DRIVE?時，打入

1

表示磁碟機的代號（注意，本例是在單一磁碟機下操作的）。

7. 現在，您準備回答那些檔案希望轉換格式：FILENAME?.
打入

=

此等號(=)是代表所有13磁區格式的檔案名稱;它是一個通配字元(Wildcard)。此處相當於轉換整片13磁區格式的磁碟。

8. 在開始轉換之前，CONVERT13會詢問是否需要"複查"，請您回答DO YOU WANT PROMPTING? 問題，現在，打入N（表示No，不需要）

9. 螢幕顯示 INSERT DISK(S) THEN PRESS <ESC> TO RETURN TO MAIN MENU OR ANY OTHER KEY TO BEGIN此段訊息時，自磁碟機取走 **SYSTEM MASTER** 磁碟。從現在開始，您只是在您的 13 磁區磁碟和一片已啓始化磁碟之間操作。

10. 將 13 磁區磁碟插入磁碟機內。遵循 INSERT SOURCE DISK AND PRESS A KEY的提示，按下 RETURN 鍵。現在，CONVERT13 程式開始尋找 13 磁區磁碟內第一個檔案，並讀取該檔案內容。

11. CONVERT13 顯示INSERT DESTINATION DISK AND PRESS ANY KEY訊息，表示它等待您插入 16 磁區磁碟，並按下 RETURN 鍵。

12. CONVERT13 在過程中會反覆地指示您插入原始磁碟（即您的 13 磁區磁碟）， 讀取檔案內容，再請您插入目的磁碟（即已啓始化磁碟），直至每一個 13 磁區格式的檔案已轉換完成並存入 16 磁區格式的磁碟。最後，會顯示 DONE 字樣，表示已完成所有轉換。注意，在讀取一較長的檔案情況下，您可能需要抽換磁碟數次，才能完成完整的檔案轉換。

兩部磁碟機：假若您有一部以上的磁碟機，可在CONVERT13 詢問原始和目的磁碟所在的擴充槽編號及磁碟機代號時，打入對應的數值。在開始轉換之前，務必將磁碟放在正確的磁碟機內。

當螢幕顯示

```
INSERT DISK(S) THEN PRESS <ESC> TO RETURN TO
MAIN MENU OR ANY OTHER KEY TO BEGIN
```

訊息時，您尚有機會取消轉換檔案的念頭；假若您按下 ESC，CONVERT13 停止執行，並回復功能表。

■假若您希望轉換的某一個 13磁區格式之檔案，在目的磁碟也有一相同檔名的檔案，則螢幕會顯示

```
FILE [filename]
ALREADY EXISTS.
TYPE IN A NEW FILE NAME FOR THE COPY
OR <RETURN> TO REPLACE EXISTING FILE
OR <CONTROL-C><RETURN> TO CANCEL COPY:
```

您可以打入一個新檔名，或者直接按 RETURN 表示轉換該13磁區格式的檔案，並取代目前的16 磁區格式的檔案，也可以按 CONTROL-C RETURN 中止轉換工作。

A.1.2 通配字元

在上列的操作例中，您已經使用〃通配字元〃（=）來表示磁碟內的所有檔案。若在檔案名稱中使用了它，可以代表任何字元，或一組任何字元。例如，在回應 FILENAME? 問題時，打入 FI=LE，即表示轉換檔名以 FI 起始且以 LE 結尾的所有檔案。同理，=TEXT，即代表檔名以TEXT結尾的所有檔案，而 =*= 則代表檔名中含有星號的所有檔案。

當您使用了通配字元，CONVERT13會詢問 DO YOU WANT PROMPTING ? 若您打入Y，表示CONVERT13 磁區磁碟上找到合乎檔名規格的檔案時，會暫時停下來，再次詢問確認一下，它是否即是您希望轉換的檔案。若不需要〃複查〃，則打入N，CONVERT13即直接進行合乎檔名規格的所有檔案轉換工作，不再暫停以便再次確認。若打入Q，則立刻回復功能表畫面。

A.2 使用未轉換的13磁區磁碟

假若您的13磁區磁碟被加上了"拷貝保護",無法進行轉換,您仍然可在16磁區格式的系統中使用它。在SYSTEM MASTER磁碟上執行START13程式,或者以BASICS磁碟來啓動系統。

BASICS 磁碟的功能完全與START13程式相通,差異點是您必須用BASICS磁碟本身來啓動它,而RUN指令是在SYSTEM MASTER磁碟直接執行START13程式。BASICS磁碟內含有兩種BASIC語言,但它已不再是標準程式集(package)的一部份。

A.2.1 使用START13 程式

START13是在SYSTEM MASTER磁碟內的一個特別載入程式(loader program),它自16磁區磁碟讀取其內含資訊,存入記憶體內。它同時也允許您寫入和讀取16磁區檔案內容。順便一提,START13其功能是相當於早先公佈的BOOT13程式(以BRUN指令執行之),亦即是RUN START13的效果相當於BRUN BOOT13。

A.2.1.1 範例

下列的START13程式操作例是假設您有一部磁碟機,磁碟機控制界面卡插在編號6的擴充槽。

1. SYSTEM MASTER磁碟放在第一部磁碟機,打入下列指令執行START13程式,

RUN START13

按下 RETURN 鍵後， EXECUTING BOOT13 訊息立即顯示於螢幕上，隨後，出現圖A－3的功能表。記住，START13執行BOOT13。

2. 在螢幕的頂端，出現SLOT TO BOOT FROM(DEFAULT = 6）?的問句，詢問13磁區磁碟（原始磁碟）所在的界面卡擴充槽編號。將您的13 磁區磁碟放入第一部磁碟機並按下 RETURN 鍵。

假若您的13 磁區磁碟是屬於"轉鑰磁碟"（turnkey disk），則某一程式自動執行之。假如該磁碟含有Apple Writer 1.0，則隨即顯示Apple Writer的指令功能表畫面。

```
13-SECTOR BOOT UTILITY

COPYRIGHT APPLE COMPUTER,INC. 1979
```

圖A－3 START13標題畫面

A.2.2 使用BASICS磁碟

BASICS磁碟也能夠載入13磁區磁碟的內含資訊至記憶體，同時也允許您存取（讀取和寫入）13 磁區格式的檔案。要使用BASICS 磁碟，請放入第一部磁碟機，而且其界面卡必須放在編號6的擴充槽上。當螢幕顯示如下的訊息

```
APPLE II

INSERT YOUR 13-SECTOR DISKETTE
AND PRESS RETURN
```

請插入任何一片 13 磁區磁碟，並按下 RETURN 。(附註：BASICS 磁碟本身並未包含DOS)

啓動BASICS 磁碟：在早期的 Apple II 有關手册裡"啓動 BASICS 磁碟"動作，現在您可以 BASICS 磁碟或執行SYSTEM MASTER 磁碟內的START13 程式取代它。

附錄B

儲存資訊的內部處理過程

本附錄討論磁軌（track）和磁區（sector）之間的關係，並詳細描述如卷册（volume）的“目次表”之類的儲存基本要素。同時也解釋機器語言程式設計者如何直接使用磁軌和磁區以及可重定址的 **DOS** 的副程式。

B.1 磁軌與磁區

16 磁區的 **DOS** 系統中，資訊記錄於磁碟上，係分佈於 35 個同心環帶，此環帶稱爲**磁軌**（track）。磁軌編號由最外環的第 \$00軌到最內環的第 \$22 軌、當磁碟旋轉時，磁碟機的磁頭（recording and reading head）移進或移出以便接近各個不同的磁軌。

磁碟上每個磁軌再分成 16 個**磁區**（sector）編號由 \$0 到 \$F。當磁碟機的磁頭停在某一圈磁軌上時，每當磁碟片轉動，磁軌上的 16 個磁區即可依序通過磁頭。

每塊磁區保存 256 個（\$100）位元組的資訊。如欲儲存資訊到

註：**錢號**（dollar sign, \$）表示十六進位數字。

磁碟上，DOS先將資訊暫時放入足夠容納一塊磁區的記憶區域，稱爲**檔案保留區**（file buffer）。當檔案保留區塡滿之後，DOS將資訊存入磁碟中一塊磁區。然後 DOS 將保留區塡零，等待爾後接下去塡256位元組的資訊。

DOS 總是將資訊以 256個位元組爲單位存入磁碟，每次一定塡滿一塊磁區。如果檔案保留區少於 256個位元組，則其餘位元組全塡上零。

只要能在磁碟上找到一塊未使用的磁區，DOS即可開始儲存一個程式或文字檔。當一個磁區塡滿時，DOS繼續尋找其他空出的磁區，也許在其他磁軌上可找到，然後再將資訊塡入。此過程反覆執行，直到整個檔案全部存完爲止。

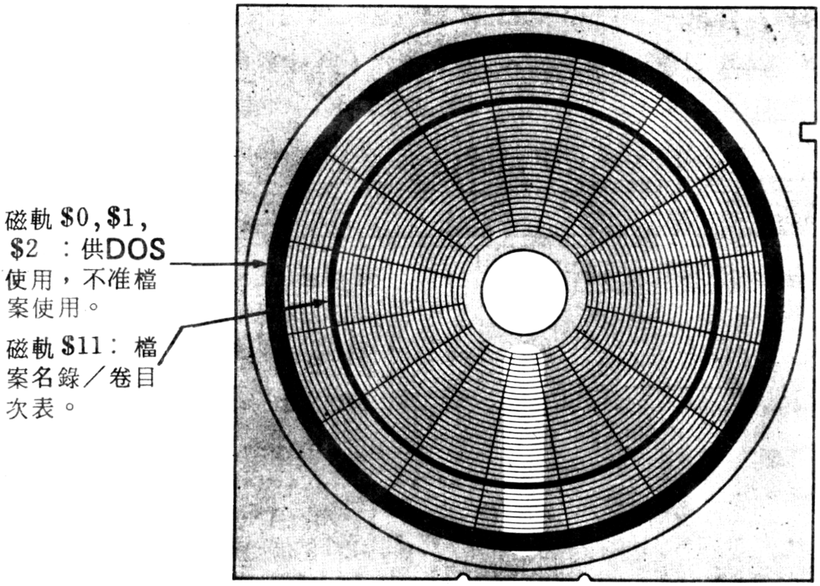
爲了要追踪檔案使用了那些儲存位址，DOS保持一份各磁軌與磁區的使用表，記錄檔案使用那些區域。然後，DOS將“**磁軌／磁區表**”（ track / sector list ）存入磁碟其他可用的磁區。

順便一提：當已存在的檔案要使用時，“磁軌／磁區表”讀入檔案保留區的第 2 塊 256位元組區域。

最後，檔案名稱、檔案型態、使用的磁區數和檔案的磁軌／磁區表的磁碟位址，全都記錄在一塊特殊區域號 $11軌，稱爲**檔案名錄**（directory）。這時，DOS修改磁碟的**卷目次表**（volume table of content），以便顯現各磁軌有那些磁區正在使用中。

註：檔案保留區（ file buffer ）=595位元組：256 個儲存資訊，256個是“磁軌／磁區表”，83 個供記錄檔案使用狀況之用。

註：卷目次表（ VTOC ）亦可稱爲“磁碟內容總表”，因爲 Volume 係檔案的集合體，存在於「磁碟」上，VTOC 即追踪記錄目前各檔案的狀況，及反應磁碟的可用空間狀況的一表格。



圖B-1　說明磁碟的磁區結構

B.2　檔案磁區的內容

本節描述各個儲存基本要素的格式：檔案磁區（file sector）、磁軌／磁區表（track／sector list）、磁碟的檔案名錄（disk directory）、卷目次表（volume table of content）和磁軌配置圖（track bit map）。

B.2.1　各類檔案型態所用的檔案磁區格式

所有資訊，無論其檔案型態，都以16進位位元組來儲存。但是DOS根據不同的檔案型態：A、I、T、B或R，以不同的方式來

解釋其資訊。

表B-1是對應於不同的檔案型態，其檔案磁區的格式。

表B-1 不同檔案型態其檔案磁區之格式

檔案型態	磁 區	位元組	位元組內容
A，I	第一磁區	$0	程式長度，低位元組。
		$1	程式長度，高位元組。
		$2至$FF	符號化程式(Tokenized program)
	後續的其他磁區	所有位元組	符號化程式
T	所有磁區	所有位元組	文字的ASCII 碼，最高位元是1。每個字元佔一位元組($00表示檔案結束)
B	第一磁區	$0	RAM起始位址，低位元組。
		$1	RAM起始位址，高位元組。
		$2	RAM映像長度，低位元組。
		$3	RAM映像長度，高位元組。
		$4至$FF	二進制資料
	後續的其他磁區	所有位元組	二進制資料

註：檔案型態(file type)是檔案保留區最後83位元組使用狀況資訊中的一部份。

註：有關檔案型態R的磁區格式，請參考6502 Assembler/DOS Tool Kit。RAM"映像"(RAM image)亦即是存於RAM的"實際內容"。

B.2.2 磁軌/磁區表(track/ sector list)

當檔案存入磁碟時，DOS產生一份記錄檔案使用狀況的磁碟位址表。**磁軌／磁區表** 存入磁碟的方式，與檔案本身相同。表B-2說明磁軌／磁區表的第一磁區之內容。每個檔案都分配"一對"位元組表示磁碟位址，其中一位元組是磁軌，另一者是磁區位址，此對位元組稱爲"磁軌／磁區對"(track／sector pair)。請注意位址$1和$2 的"**環結對**"(**link pair**)，這兩個位元組存有下一個磁軌／磁區表的位址。

當此表超出122磁軌／磁區對時，下一個的磁軌／磁區表的結構與此表相同，各個磁軌／磁區對的內容是隨後的122個檔案磁區的位址(在第幾磁軌第幾磁區)。此外，第1和第2位元組的環結對也不相同(假若磁軌編號相同，表示前後兩個磁軌／磁區表是在同一磁軌上。)，當環結對的兩個位元組都是零時，表示該表是一連串的磁軌／磁區表的最後一個表格。

表B-2 磁軌／磁區表的第1磁區

位 元 組	位元組的內容
$0	不用
$1	環結：下一個磁軌／磁區表的磁軌編號。
$2	環結：下一個磁軌／磁區表的磁區編號。(假若這兩個位元組都是0，表示"已無下一個表"(no link))。
$3至$4	不用
$5至$6	本磁軌／磁區表的第一個檔案磁區，在檔案的排列編號(即相對於檔案開頭處，是"第幾個"檔案磁區。)。
$7至$8	不用

$C	第一個檔案磁區的磁軌編號
$D	第一個檔案磁區的磁區編號
$E	第二個檔案磁區的磁軌編號
$F	第二個檔案磁區的磁區編號
⋮	
$FE	第122個檔案磁區的磁軌編號
$FF	第122個檔案磁區的磁區編號

任何磁軌／磁區對，如果是0/0表示沒有指定任何磁區，通常即表示“檔案結束”。但是隨意存取文字檔內可以有很多磁區指定爲0/0，稍後的任何時候再寫入資訊。

隨意存取檔中，只有眞正存有資訊的磁區，其磁軌／磁區表上對應的“磁軌／磁區對”才不是0/0。DOS會計算磁軌／磁區對在表上的正確位置，沒有使用的磁區其對應的“磁軌／磁區對”全部塡零。

隨意存取檔中，當您指定資料錄長度爲127（即相當於每個磁區可放兩個資料錄），而且只有第2700資料錄寫入資料，則DOS共使用13個磁區：一個存錄號2700的內容，另外12個存12個磁軌／磁區表。從錄號0到2683的資料錄其內容可能要佔去1342個磁區，但是必須等到這些資料錄寫入資訊，才會眞正佔用磁碟空間。磁軌／磁區表中，保存資料錄0至2683的磁區之位址總共用掉11個磁區。（譯註：每一磁軌／磁區表可保留122檔案磁區，即122＊2＝244資料錄，11＊244＝2684個資料錄）

B.2.3 磁碟檔案名錄（disk directory）

對於每一片已啓始的磁碟，DOS都保留第$11磁軌做爲磁碟檔案名錄（disk directory），保存磁碟中各個檔案有關的資訊：檔案

名稱（file name）、檔案型態（file type）、檔案使用的磁區數（MOD 256）、檔案是否鎖住和檔案的第一個磁軌／磁區表位址。**CATALOG** 指令顯示所有這些資訊，只有磁軌／磁區表的位址沒有列出。

表 B-3 列出磁碟檔案名錄的每一個磁區之內容。

表 B-3　磁碟檔案名錄的一個磁區

位元組	位元組內容
\$0	不用
\$1	環結：下一個檔案名錄的磁軌編號（通常是 \$11）
\$2	環結：下一個檔案名錄的磁區編號。 （如果環結的兩位元組皆爲 0，則沒有環結。）
\$3 至 \$A	不用
\$B 至 \$2D	檔案 1 的名錄項目（directory entry）
\$2E 至 \$50	檔案 2 的名錄項目（directory entry）
\$51 至 \$73	檔案 3 的名錄項目（directory entry）
\$74 至 \$96	檔案 4 的名錄項目（directory entry）
\$97 至 \$B9	檔案 5 的名錄項目（directory entry）
\$BA 至 \$DC	檔案 6 的名錄項目（directory entry）
\$DD 至 \$FF	檔案 7 的名錄項目（directory entry）

表 B-3 中，檔案名錄的 7 個檔案之編號是隨意編的。如果刪除某一檔案，**DOS** 會標記此名錄項目已刪除，爾後如果有一檔案新建，則 **DOS** 用已標記的名錄項目來儲存新檔案的項目。換句話說，**DOS** 原先以所示的順序將檔案塡入名錄中，只要有某一個檔案被刪除，其順序就失去意義。

註：檔案名錄的細項內容，請參考 B-4。

檔案名錄由第$11磁軌、第 $F磁區開始存放。這個起始磁區會出現在“卷目次表”的第一、第二位元組。當需要更多的空間存放更多的名錄項目時，第 $F 磁區連結到 $E；如果還再需要空間，則連結到$D ；餘此類推，直到$1 。因此，檔案名錄允許最大儲存105個不同檔案項目。

每個檔案項目的格式如表B-4所示。

表 B-4　一個檔案其名錄中項目的內容

相對的位元組	位元組內容
$0	檔案內“磁軌／磁區表”的磁軌編號。（如果檔案被刪除，則原來的值抄到$20，同時第 $0 位元組改成$FF）。
$1	“磁軌／磁區表”的磁區編號
$2	檔案型態
$3 至$20	檔案名稱
$21	磁區數：檔案所佔用的磁區個數（MOD 256）
$22	結束符號（$00）

檔案名錄中項目的“相對位元組$0”是該項目的一個“參考點”，標示$2即表示在“該項目內”是第 $2 個位元組。實際上各項目是以不同的位元組編號起頭。要找出對應於相對位元組的“絕對”磁區位元組數，只要將相對位元組號加上項目的第一個絕對磁區位元組號即可得。（如表B-3所示，例如第3項目的“檔案型態”其絕對位址是$51+$2=$53）。

因爲只用一個位元組來記錄檔案磁區數，最大的磁區數是255（$FF）。當檔案超過255個磁區時，磁區數（即是CATALOG指令所顯示的）又從第0磁區重頭算起。這不會影響到檔案的使用，但是可能使您對磁碟的剩餘空間有錯誤的印象。

“檔案型態”此位元組（在表 B-4 中，檔案的名錄項目之“相對位元組 2 ”）的 8 個位元（bit）。其意義如表 B-5：

表 B-5　“檔案型態”位元組的格式

位　　元	CATALOG 符　　號	指定的檔案型式
7	*	如果此位元是 1，檔案鎖住（防止寫入）。 如果此位元是 0，檔案未鎖定（沒有寫入保護）。
6 — 5		供以後擴充其他檔案型態（通常是 0）
4	R	此位元是 1 時，即是可重定址的 EDSAM 檔案。
3	S	供以後擴充其他檔案型態（通常是 0）
2	B	此位元是 1 時，二進制機器碼檔案（binary file）。
1	A	此位元是 1 時，Applesoft BASIC 檔案。
0	I	此位元是 1 時，Integer BASIC 檔案。
6—0	T	當位元 6 至 0 全是 0 時，文字檔。

檔案型態係由位元 6 至位元 0 之間的任何一個位元來決定的。如果六個位元全是 0，則預設此檔爲文字檔。

指定檔案型態的位元組，其值如下表 B-6 所列：

表 B-6 “鎖住／解鎖”檔案型態的對應值

檔案型態	檔案型態的對應值 檔案鎖住 (lock)	檔案未鎖 (unlocked)
T	$0	$80
I	$1	$81
A	$2	$82
B	$4	$84
S	$8	$88
R	$10	$90

B.2.4 卷目次表 (Volume Content of Table)

第 $11 磁軌的第 $0 磁區存有磁碟的“卷目次表”(volume table of content ，**VTOC**)。卷目次表所存的資訊，如表 8-7 所示。

表 B-7 卷目次表 (第$11 磁軌，第 $0 磁區)

位元組	值	說明
$0	$2	可變值；不用。
$1	$11	第一個名錄磁區之磁軌編號。
$2	$0F	第一個名錄磁區之磁軌編號。
$3	$3	DOS 的發行號碼 (版本號碼)
$4	$0	不用
$5	$0	不用
$6	$1－$FE	磁碟卷號 (預設值：$FE)

註：VTOC＝volume table of content (卷目次表)

$7－$26	$0	不用
$27	$7A	“磁軌／磁區表”內各“磁軌／磁區對”的可能最大值。
$28－$2F	$0	不用
$30	$FF	使用中的最後一個磁軌。
$31	$FF	使用的方向（＋表示更高磁軌）。
$32	$00	不用
$33	$00	不用
$34	$23	每片磁碟的磁軌數。
$35	$0F	每個磁軌的磁區數。
$36	$00	每個磁區的位元組數，低位元組。
$37	$01	每個磁區的位元組數，高位元組。
$38－$3B	$0	第0磁軌的位元配置表*（不能使用）。
$3C－$3F	$0	第1磁軌的位元配置表*（只有VTOC改變時，才能使用）。
$40－$43	$0	第2磁軌的位元配置表*
$44和$45	?	第3磁軌的位元配置表*
$46和$47	$0	第3磁軌的位元配置表*
$48和$49	?	第4磁軌的位元配置表*
$4A和$4B	$0	第4磁軌的位元配置表（未用）
⋮		（bit map,位元配置表）
$78和$79	?	第10磁軌的位元配置表*
$7A和$7B	$0	第10磁軌的位元配置表*
$7C－$7D	$0	第11磁軌的位元配置表*（檔案名錄與VTOC）
$7E－$7F	$0	第11磁軌的位元配置表*（未用）
$80和$81	?	第12磁軌的位元配置表*（未用）
$82和$83	$0	第12磁軌的位元配置表*
$C0和$C1	?	第22磁軌的位元配置表*
$C2和$C3	$0	第22磁軌的位元配置表*（未用）
$C4－$FF	$0	不用

(＊譯註：此處的"磁軌的位元配置表"是指該磁軌 16 個磁區的配置圖，以 2 個位元組（共 16 個位元）的每一個位元代表某一磁區。)

B.2.5 磁軌的位元配置表（track bit map）

VTOC（表B-7）表第$38位元組開始的一系列"連續的 4 位元組"組合成磁碟中 35 個磁軌的"**磁軌位元配置表**"（track bit map)。磁軌配置表中各位元（對應至各個磁區），以 1 代表某磁區正使用中，以 0 表示某磁區尚未被使用。每個磁軌的位元配置表採用如表B-8 的格式。（譯註：由於 **DOS** 3.3 是16磁區格式，僅用前二個位元組表示 16 個磁區，後二個位元組則保留給未來的 **DOS** 版本使用。）

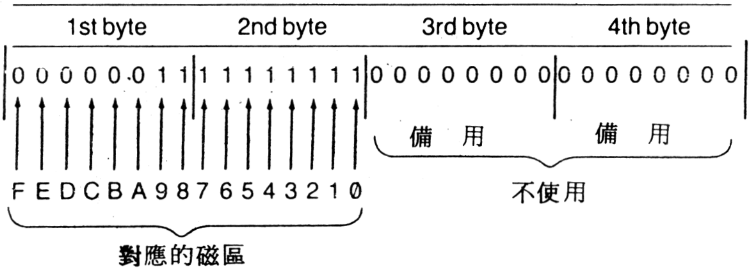
表B-8 磁碟片的磁軌位元配置表

位元組	位元	指定磁區	位元組	位元	指定磁區
1	7	$F	2	7	$7
	6	$E		6	$6
	5	$D		5	$5
	4	$C		4	$4
	3	$B		3	$3
	2	$A		2	$2
	1	$9		1	$1
	0	$8		0	$0
			3 和 4	所有位元	備用

當磁軌位元配置表的某一位元是1時，此位元所對應的磁區沒有被使用。當配置表中有一位元是0時，其對應的磁區正使用中。標定爲“備用”的位元組全部塡0，這些位元備而不用。

表B-9說明標準的“磁軌的位元配置表”，其磁區編號皆是十六進位。

表 B-9 標準磁軌位元配置表



1 = 可用磁區（VTOC 表第$30,$31 也是1）
0 = 已使用的磁區

當您利用WRITE、SAVE或BSAVE等指令來儲存檔案時，則儘可能將整個磁軌分配給此檔案使用，此時磁軌的配置表顯示整個磁軌使用中。然後，當檔案關閉，VTOC修改之後，實際上尚未使用的磁區再設成“未使用”。

釋放磁區：眞正用來存檔案資訊的磁區，只有在檔名由名錄中刪除之後，才能設成“未使用”。例如，某一磁碟片存有一個佔100磁區的BASIC檔案，檔名爲BIG。如果您保存（save）一個佔2磁區的檔案，以相同的檔名存入相同的磁碟片，您將蓋過舊的BIG檔案。但是檔案目錄仍舊將您的2磁區檔案當做100磁區檔案。如果想要釋放

此檔案中不需要的 98 個磁區，則使用如下的一串指令

```
LOAD BIG
DELETE BIG
SAVE BIG
```

對於二進制機器碼檔案（binary file）中多餘的磁區，也可以用類似的一串指令將其釋放。

如果您想保存文字檔中的資料，並釋放不需要的磁區，您必須將各資料欄一一讀入記憶體。如果您將所有資料欄存入陣列（array），即可刪除此檔案，然後再以相同的檔名將每一個資料欄寫入磁碟。

另一個釋放文字檔的磁區之方法是，將各資料欄立即寫入一個不同名的新檔案。寫完最後一個資料欄之後，將原先的檔案刪除。

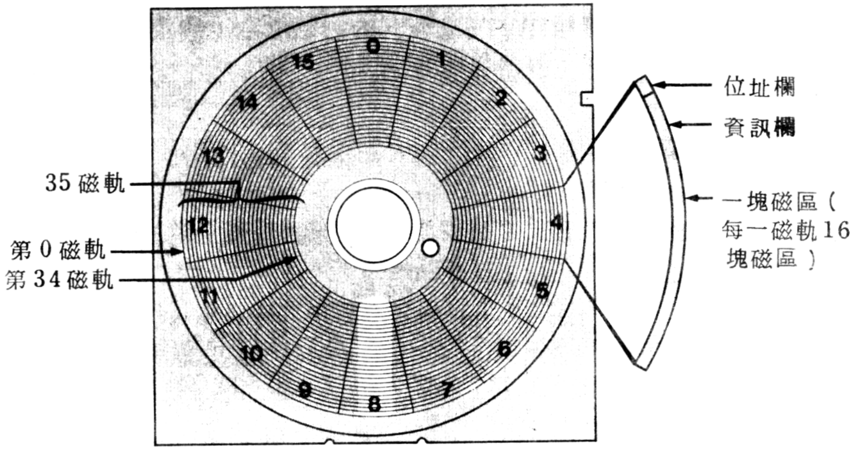
B.3 磁軌與磁區的分配

每一片磁碟含有 35 個磁軌，其中 3 個保留供 DOS 使用，1 個保留做爲檔案名錄，剩下 31 個磁軌可供利用。每個磁軌含有 16 個磁區，因此有 31×16 或 496 個磁區供您使用。

新啓始的磁碟中，磁區由第 $F 磁區開始填起，往回依次填到第 0 磁區。磁軌則由第 $12 磁軌（恰是名錄／VTOC磁軌往內一圈）開始填起，逐漸往內圈填，直到第 $22 磁軌。當第 $22 磁軌填滿之後，開始填第 $12 磁軌（名錄／VTOC磁軌往外一圈），然後逐漸往外填，直到第 3 磁軌爲止（可用的最外圈磁軌）。

B.4 爲磁軌與磁區定址：RWTS 副程式

如圖B-2，每一磁區都有一個位址欄（address field）和一個資訊欄（date field）。位址欄所存的資訊包括磁頭所在的磁軌、磁頭下一步所前往的磁區編號和磁碟的卷號。資訊欄才眞正存此磁區的256位元組資料。



圖B-2　磁軌與磁區

DOS 爲您計算磁軌和磁區，但是機器語言程式師也可以由機器語言程式呼叫 DOS 的RWTS（Read or Write a Track and Sector，讀寫一磁軌與磁區）副程式，來指定磁碟上的磁軌與磁區。

RWTS副程式由磁碟上一特定的磁軌和磁區，寫入或讀出資訊。要使用RWTS之前，必須先在記憶體中建立與保存資訊於輸出入區段（I/O block）、設備特性表（device-characteristics table）和控制用副程式。

- “輸出入區段”告訴RWTS副程式使用那一個擴充槽和那一部磁碟機，並指明卷號、要使用的磁軌與磁區以及要讀或要寫。
- “設備特性表”描述裝備，在此處爲磁碟機。
- 控制副程式（controlling subroutine）將輸出入區段的起始位址存入暫存器A和Y，其中A暫存器存此位址的高位元組，Y暫存器存其低位元組。最後，此控制副程式將執行控制權轉給RWTS副程式的起始位址（利用組合語言JSR指令），其位址是$3D9。

B.4.1 範 例

以下的例子中“輸出入區段”、“設備特性表”和控制副程式”被載入記憶體時，係由位址$C00開始。

此輸出入區段中指定第6槽的第一部磁碟機爲輸入和輸出裝置，將記憶體位址$C0A起256個位元組寫入第$12磁軌的第$06磁區。

位 址	碼	目 的
$C0A	01	輸出入區段的標示值（一定是$01）
$C0B	60	槽號乘以16
$C0C	01	磁碟機號碼
$C0D	00	期望的卷號
$C0E	12	磁軌編號
$C0F	06	磁區編號
$C10	20	設備特性表起始位址的低位元組

註： 有關“輸出入區段”，請參考B-10表。

註： 有關設備特性表，請參考表B-11。

$C11	0C	設備特性表起始位址的高位元組
$C12	00	資訊保留區起始位址的低位元組
$C13	20	資訊保留區起始位址的高位元組
$C14	00	不用
$C15	00	不用
$C16	02	指令碼，$02表示“寫入”
$C17	00	錯誤號碼
$C18	00	眞正的卷號
$C19	60	上次使用的槽號
$C1A	01	上次使用的磁碟機號碼

以下的控制副程式將輸出入區段的起始位址存入暫存器A和Y，然後執行的控制權轉入RWTS副程式。

```
$C00  A9  0C       LDA   #$0C     將$0C 載入暫存器A
$C02- A0  0A       LDY   #$0A     將$0A 載入暫存器Y
$C04- 20  D9 03    JSR   $03D9    跳至RWTS 副程式
$C07- 60           RTS
$C08- 00           BRK
```

將設備特性表存入位址$C20，接於輸出入區段之後。（在本例中，上述的輸出入區段中位址$C10和$C11所存的值，指向此起始位址）。設備特性表存有如下的碼值：

位　址	碼	目　　的
$C20	00	存設備型態的代碼
$C21	01	存每圈磁軌的相位數（phases number per track）
$C22	EF	存時間計數（time count）
$C23	D8	存時間計數

建立輸出入區段於＄C0A，設備特性表於＄C20，以及控制副程式於＄C00，然後可利用以下兩指令來執行整個程式：

```
C00G                    或                    CALL 3072
```

B.4.2 格式：輸出入區段與設備特性表

表 B-10 說明輸出入區段的格式。

表 B-10 輸出入區段之格式

位元組	名稱	目的
01	IBTYPE	指明此段爲輸出入區段（其值＄01）
02	IBSLOT	插有磁碟機控制卡的擴充槽號乘以16，例如使用第6槽則存＄60。
03	IBDRVN	所使用磁碟機號碼（＄0或＄1）
04	IBVOL	所使用的卷號，＄00則不核對卷號。
05	IBTRK	所使用的磁軌，必須介於＄00至＄22之間（0到34）。
06	IBSECT	所使用的磁區，必須介於＄00至＄0F之間（0到15）。
07－08	IBBUFP	RWTS所使用的資訊保留區（記憶體中256個位元組）之起始位址，資訊由保留區寫入磁碟。當您由磁碟讀入資料時，資訊先放入保留區。RWTS只讀整塊256位元組的資料。
0B－0C		不用
0D	IBCMD	指定RWTS所要執行的動作。可存入＄0D(13)位址的值有： ＄00 一啓動磁碟機，將磁頭定位。

		$01 一將指定磁軌與磁區內的256位元組讀入資訊保留區。 $02 一將保留區 256 位元組寫入指定磁軌的磁區。 $04 一將磁碟片格式化（format），每磁軌與磁區寫入“同步用 4 位元”（self-synchronizing nibbles）。因整片磁碟已格式化，所以不管位元組 $05 和 $06 所存的值。整片格式化的磁碟即可使用。直到您或 DOS 放入資料，磁碟才存有資料。
0F	IBSTAT	記錄錯誤的編號： $00 一無誤 $10 一磁碟處於寫入保護狀態 $20 一卷號，與位元組 $04所指定的不同。 $40 一磁碟機不正常運作
0F	IBSMOD	眞正的卷號
10	IOBPSN	上次使用的槽數乘以 16，例如，如果上次您使用第 5 槽，則存 $50。如果沒有控制卡插於此槽，Apple II會“當掉”。
11	IOBPDN	上次使用的磁碟機代號，$01 或 $02。

表 B-11說明設備特性表的格式。

表 B-11 設備特性表之格式

位元組	名稱	目的
01	DEVTPC	設備型態碼，指明使用設備之類型。如果是磁碟機，此位元組存 $ 00。
02	PPTC	此處存 $ 01。
03-04	MONTC	以 100 微秒(micro-second)爲單位，存入馬達起動時間值的補數(complement)。如果此設備是磁碟機，將 $ EF 放入第 3 位元組，將 $D8 放入第 4 位元組。

B.5 第3分頁的DOS向量

DOS佔用 10.5 K的RAM。此外，DOS在第 3 分頁(Page 3由 $ 3D0 到 $ 3FF，在第 1 個主要文字分頁之下)使用一群向量(vector)。

因爲 DOS 能依照不同大小記憶體的系統或者是否來自主磁碟(master disk)等不同情況，而將 DOS 置於記憶體不同的區域，如此可呼叫的 DOS 副程式其位址會因不同的系統和不同的磁碟片而有改變。如果您將 DOS 副程式的位址放入第 3 分頁固定位址的某一向量中，則不必擔心 DOS 的副程式在不同的記憶體位址。

對於 DOS 3.3 以上的版本，也可以利用第 3 分頁中的位址來定出 DOS 副程式的位址。DOS較早的版本利用位址 $ 300 到$ 3CF 來載入BOOT1 程式；DOS 3.3 則利用此位址當做資訊保留區(data buffer)和磁碟碼翻譯表(disk code translate table)。

註：DOS和兩種 BASIC 所使用的區域，請參考圖 6-3。

註：表 B-13 列出DOS 3.3 所使用的第 3 分頁位址及其內容。

表B-12　DOS 3.3 的向量（$3D0－$3FF）

位　　址	內　　　容
$3D0	JMP 至 DOS 溫啓動（warm-start）副程式，以便重新接通 DOS 並保持目前程式與 MAXFILES 不變。
$3D3	JMP 至 DOS 冷啓動（cold-start）副程式，重新接通 DOS，相當於重新啓動（reboot）。目前程式會消失不見，MAXFILES 恢復預設值，HIMEM 重設。
$3D6	JMP 至 DOS 的檔案管理副程式。因此您的組合語言程式可以呼叫它。
$3D9	JMP 至 RWTS 副程式。因此您的組合語言程式可以呼叫它。
$3DC	此副程式定出"輸入參數表"（input parameter list）的位址。使得您的程式在使用檔案管理程式之前，可先建立此表。
$3E3	此副程式定出"IOB 參數表"（IOB parameter list）。使得您的程式呼叫RWTS之前，能建立此表。
$3EA	JMP至一個 DOS 副程式，重新建 DOS 的 I/O向量（vector）。
$3EF	JMP至一個副程式，以便處理 BRK 機器指令（僅 Autostart ROM適用）。通常此向量所指到的副程式可以顯示暫存器的內含值。
$3F2-$3F3	存一個副程式的高位元組與低位元組的位址，此程式可以處理 Autostart ROM的RESET。通常保存 DOS 的重新啓動之位址。
$3F4	這是冷啓動（cold-start）位元組，存有 RESET 位址的補數（complement），以便區分冷啓動與溫啓動。如果冷啓動發生時，Autostart ROM 不管 $3F2 所存的位址，並試圖重新啓動磁碟。要防止此種情況發生，可以更改 $3F2 的內容以處理您自己的重設動作（reset），將新值與 $A5 存入做為EOR，然後將結果存入冷啓動位元組。

$3F5	JMP至一個機器語言副程式。當Applesoft 指述使用 & (ampersand) 符號時，就會呼叫此副程式。
$3F8	JMP至一個機器語言副程式。當監督程式讀入 CONTROL - Y 時，會呼叫此副程式。
$3FB	JMP至一個機器語言副程式。當"無法遮蔽的中斷" (nonmaskable interrupt)情況發生時，ROM會呼叫此程式。
$3FF- $3FF	存一副程式的位址之低位元組與高位元組。當"可遮蔽中斷" (maskable interrupt) 情況發生時，ROM呼叫此程式。

B.6 第0分頁的使用情形

第0分頁 (zero page) 由位址$0到$FF，是監督程式和 BASIC 的使用空間。機器語言程式設計者可利用間接索引指令 (indexed indirect command) 來使用此塊記憶區中的特殊功能之位址，以確保程式的精簡。表 B-13說明DOS如何使用第0分頁。

表 B－13 DOS 使用第 0 分頁的情形

Byte	Use
$24	Cursor horizontal (DOS)
$26-$27	Read buffer used by boot routine
	RWTS workspace
$2A	RWTS workspace
$2B	Boot slot times 16
	RWTS workspace
$2C	Checksum
$2D	Sector number (RWTS)
$2E	Track number (RWTS)
$2F	Volume number (RWTS)
	RWTS workspace
$33	Prompt character (DOS)
$35	Drive number in high bit (RWTS)
$36-$37	CSWL,CSWH (DOS)
$38-$39	KSWL,KSWH (DOS)
$3C	RWTS workspace
	Device-characteristics table address (RWTS)
$3D	Sector number (BOOT)
	Device-characteristics table address (RWTS)
$3E-$3F	Address of ROM sector-read subroutine (BOOT)
	RWTS buffer address
$40-$41	DOS image address (BOOT)
	DOS file buffer address
$41	Volume number used in INIT
$42-$43	DOS buffer address
$44	Track number used in INIT
$45	Sync byte used in INIT
$46-$47	RWTS workspace
$48-$49	IOB address pointer (RWTS)
$4A-$4B	LOMEM address for Integer BASIC (DOS)
$4C-$4D	HIMEM address for Integer BASIC (DOS)
$67-$68	Pointer to beginning of Applesoft program
$69-$6A	Pointer to start of Applesoft variable space
$6F-$70	Pointer to start of Applesoft string storage
$73-$74	Highest location in memory, plus one, for Applesoft
$76	Applesoft line number being executed
$AF-$B0	Pointer to end of Applesoft program
$CA-$CB	Pointer to beginning of Integer BASIC program
$CC-$CD	Pointer to end of Integer BASIC variable space
$D6	Applesoft write-protect flag (DOS)
$D8-$D9	Integer BASIC line number (DOS)
	Applesoft ONERR (DOS)

附錄C

DOS錯誤訊息

錯誤訊息（ error message ）警告您有一問題已發生， 指示您所下的指令不符合 DOS 的規則。錯誤訊息的格式告訴您是否由 DOS 發出，或者是由 BASIC 語言所發出。

格　　式	訊息的來源
SYNTAX ERROR	DOS
?SYNTAX ERROR	Applesoft
*** SYNTAX ERR	Integer BASIC

本附錄只討論 DOS 的錯誤，有關Applesoft 和 Integer 、 BASIC 的錯誤訊息請參考各自的手冊。

在 Applesoft ONERR副程式中，錯誤（ error ）通常以編號 1 到 15 的數目來表示。

C.1 節列出 DOS 的錯誤訊息。C.2 節討論每一個錯誤訊息，解釋各個錯誤的可能原因，並建議修正錯誤的方法。

C.1 DOS錯誤訊息

表C－1以字母順序列出 DOS 的錯誤訊息；表C－2則以錯誤號碼順序列出。

表C－1 DOS的錯誤訊息

訊息	錯誤碼	意義
DISK FULL	9	檔案目錄中檔案太多或磁碟的所有磁區已用盡。
END OF DATA	5	讀資料超出檔案末尾。
FILE LOCKED	10	試圖改寫（overwrite）、刪除（delete）或改名（rename）已鎖住的檔案。
FILE NOT FOUND	6	檔名拼錯或檔案不在磁碟內。
FILE TYPE MISMATCH	13	磁碟檔不符合指令所要的型態。
I/O ERROR	8	磁碟機門未關好、磁碟沒有啓始（initialized)、磁碟機故障或壞的磁碟片。
LANGUAGE NOT AVAILABLE	1	所要的語言不在電腦中。
NO BUFFERS AVAILABLE	12	已開啓（open）的文字檔太多。
NOT DIRECT COMMAND	15	必須在程式中使用該指令。
PROGRAM TOO LARGE	14	可用的記憶體不足。
RANGE ERROR	2,3	指令的引數值太大或太小。
SYNTAX ERROR	11	檔名、引數或逗號不合法。
VOLUME MISMATCH	7	卷號引數錯誤。
WRITE PROTECTED	4	磁碟貼有防寫貼紙。

表C－2　**DOS** 的錯誤碼

錯誤碼	訊息	意義
1	LANGUAGE NOT AVAILABLF	所要的語言不在系統中。
2,3	RANGE ERROR	指令的引數值太大或太小。
4	WRITE PROTECTED	磁碟貼有防寫貼紙。
5	END OF DATA	讀資料超出檔案末尾。
6	FILE NOT FOUND	檔名拼錯或檔案不在磁碟內。
7	VOLUME MISMATCH	卷號參數錯誤。
8	I/O ERROR	磁碟機門未關或磁碟尚未啓始化。
9	DISK FULL	磁碟片檔案太多。
10	FILE LOCKED	試圖改寫鎖住的檔案。
11	SYNTAX ERROR	檔案、引數或逗號不合法。
12	NO BUFFERS AVAILABLE	開啓的文字檔太多。
13	FILE TYPE MISMATCH	磁碟檔不符合指令所要的型態。
14	PROGRAM TOO LARGE	可用的記憶體不足。
15	NOT DIRECT COMMAND	必須在程式中使用該指令。

C.2　修正錯誤

以下的 **DOS** 錯誤訊息，以錯誤碼的順序，一一說明。

1　LANGUAGE NOT AVAILABLE

執行下列的指令FP、INT、LOAD、RUN會開始尋找適用的語言系統， LANGUAGE NOT AVAILABLE 表示 **DOS** 無法找到您的程

式所要的 BASIC —— Applesoft 或 Integer BASIC。

當您下達一指令，希望由目前磁碟機上的磁碟請求使用Applesoft 語言，而磁碟並無 APPLESOFT 或 FPBASIC 程式，這時會出現此錯誤訊息。請抽出原有的磁碟片，換上存有上述程式的磁碟片（例如， SYSTEM MASTER 磁碟片），然後再下指令。

當您下指令要求使用 Integer BASIC，而 Integer BASIC 不在電腦時也會出現此訊息。假若您的電腦有足夠的記憶體容納 Applesoft 和 Integer BASIC，您可以由 SYSTEM MASTER 磁碟片將 Integer BASIC 載入記憶體。將存有 FPBASIC 、INTBASIC 、HELLO 和 LOADER. OBJ0 的磁碟片放入磁碟機，然後再下指令。 SYSTEM MASTER 磁碟存有這些程式。

當您要求使用Applesoft, DOS 在 ROM 中找出此語言（此處的 ROM 即主機的唯讀記憶體（ read-only memory）或語言界面卡上的唯讀記憶體。如果 Applesoft 不在 ROM 中，DOS 在 RAM 中找此語言（ RAM 即隨意存取記憶體（ randon access memory））。當 Applesoft 不在 RAM 中， DOS 在目前磁碟機（依照最近使用的槽號與磁碟機代號引數）中的磁碟片上尋找此語言程式（譯註：“語言程式”即是習稱的 Interpreter ，“直譯程式”)。

當您要求使用 Integer BASIC 時， DOS 在 ROM 或 RAM 中尋找此語言。

如果您當初利用 SYSTEM MASTER 來啓動 DOS ，而您卻得到 LANGUAGE NOT AVAILABLE 的訊息，這表示您的電腦沒有足夠的空間（ 64K 或更多）供兩種語言使用。

註：唯讀記憶體（ read-only memory ）或 ROM 的內容只能讀不能再寫入。在製造過程中，資訊寫入 ROM 之後，就永遠存在，即使電腦關掉電源也不受影響。

隨意存取記憶體（ random-access memory ）或RAM的內容可以任意讀出與寫入。當電源關掉之後， RAM 的內容即消失。

2 or 3　**RANGE ERROR**

RANGE ERROR 表示指令引數的值太大或太小。表B－3說明那些引數可使用那些值。

表C－3　各引數的最小值和最大值

		引數	最小值	最大值
所有檔案：	擴充槽(Slot)	S	1	7
	磁碟機(Drive)	D	1	2
	卷號(Volume)	V	0	254
循序存取	位元組(Byte)	B	0	32767
文字檔：	相對欄(Relative Field)	R	0	32767
	絕對欄(Absolute Field，EXEC的Rn)	R	0	32767
隨意存取	資料錄長度(Record Length)	L	1	32767
文字檔：	資料錄編號(Record Number)	R	0	32767
二進制機	起始位址	A	0	65535
器碼檔案：	位元組長度	L	0	32767
DOS指	PR#	n	0	16*
令：	IN#	n	0	16*
	MAXFILES n	n	0	16

*：如果PR#和IN#的引數是8到16，則可能產生不可預期的結果。

使用的值如果超出表B－3所指定的範圍，不一定會造成 RANGE ERROR 的訊息。任何 DOS 指令若指定的引數值若小於0或大於65535，則會出現 SYNTAX ERROR 。

4 WRITE PROTECTED

WRITE PROTECTED 表示 DOS 無法儲存（Save）、寫入（Write）、刪除（delete）磁碟的資訊。（“write-protected”表示磁碟沒有“可寫缺口”（write-enable notch）或者缺口已貼上寫入保護貼紙（write-protect tab）。

- 如果磁碟缺口黏上寫入保護貼紙，撕掉此貼紙，然後再下指令。
- 如果在執行COPY程式時出現此訊息，則很可能您將磁碟放錯磁碟機。檢查磁碟機，看看磁碟的位置是否放對。
- 選用其他的磁碟來保存您的檔案。一片磁碟如果沒有可寫缺口，則永遠無法修改或刪除資訊。

5 END OF DATA

END OF DATA 的出現是您的程式擷取資訊時，讀到檔案中沒有存放任何資料的區域。

任何超出循序存取文字檔（sequential-access text file）的最後一個資料欄（field）或超出隨意存取文字檔（random-access text file）各資料錄（record）的最後一欄位址，其位元組所存的值是0或$00，即ASCII的null字元。任何指令如果要讀取null字元，就會產生 END OF DATA 訊息。

以下的情況會出現 END OF DATA 的訊息：

- B（byte）引數值太大。在循序存取時，您所指定的位元組不得超出檔案的最後一個“歸位字元”（Carriage return）。在隨意存取時，您所指定的位元組不得超出目

前使用的資料錄最後一個“歸位字元”之位置。

記住: 檔案或資料錄的第1個位元組是“位元組0”。

- R(relative-field position,相對的檔案位址)引數，於POSITION指令中指定時其值太大。在循序存取時，指定的資料欄不要超出檔案中已存在的最後一個資料欄。在隨意存取時，指定的資料欄不得超出目前使用的資料錄中已存在最後一個資料欄。

  記住: POSITION指令在檔案中“往前”移動目前位址的指標。R項引數所指定的資料欄個數係相對於目前位址(current position)。

- R引數於EXEC指令指定時，其值太大，所指定的那一列可能已經超出指令檔(command file)的末尾列。
- R引數於READ指令所指定的是空的隨意存取資料錄。當您要由隨意存取檔中讀取某一資料錄之前，要確定您先前時候曾寫入一些資訊至該資料錄內。
- READ之前OPEN指令L引數和WRITE之前OPEN指令的L引數不一致。

在INPUT或GET指令之後，出現 END OF DATA 訊息，您的程式可能有如下的情形：

- 太多個連續的INPUT指述，或INPUT指述內有太多變數。每個INPUT或INPUT變數，都是將下個資料欄讀入電腦。
- 太多個連續的GET指述。每一個GET將下一個位元組或字元讀入電腦。

當您看到 END OF DATA 訊息時，如要避免資訊流失，最好

趕緊利用**CLOSE**指令關閉檔案。有關修正程式的方法，請參考本書正文的說明。

6 **FILE NOT FOUND**

FILE NOT FOUND 表示您所指定檔名的檔案，不在目前使用的磁碟片上。

檢查磁碟檔案目錄（catalog）：

- 您可能拼錯檔案名稱，輸入不正確的檔名或者省略了隔開檔名與引數的逗號，請仔細檢查您輸入的指令。
- 您所指定的檔案可能不在磁碟中。
- 此檔案可能無意中被刪除了。

如果您每次啓動磁碟時，都會出現 FILE NOT FOUND ，則您必須告訴 **DOS** 該磁碟的歡迎詞程式（greeting program）之檔名。如果您的磁碟片沒有任何檔案需要保留，則可以利用 **INIT** 指令啓始此磁碟。如果您已不記得磁碟中的歡迎詞程式之檔名，可以執行MASTER程式將歡迎詞程式加以更名。

7 **VOLUME MISMATCH**

VOLUME MISMATCH 表示 **DOS** 指令中的卷號（volume number）與磁碟片當初啓始時所指定的卷號不同。利用**CATALOG** 指令檢查磁碟的卷號。

8 **I/O ERROR**

I/O ERROR 表示 **DOS** 無法由磁碟中擷取或存入資訊。**DOS** 發出此訊息之前，會試 96 次。

檢查您的磁碟：

- 若磁碟機的門未關好，則將其蓋上。
- 選定的或預設的磁碟機沒有放入磁碟片。請放入磁碟片，並關上門。
- 磁碟片可能尚未啓始化，趕快利用 INIT 指令準備一片可用的磁碟。
- 可能磁碟片沒有適當地插好，小心地從磁碟機中抽出磁碟片，再重新插入。
- VERIFY 指令已指示檔案存入磁碟有誤。
- 16 個磁碟區版本的 DOS 系統意圖使用 13 磁區的磁碟，請參考附錄A。

檢查您的指令引數：

- 磁碟機代號（Dn）引數所指定的磁碟機不存在於您的系統中，現在預設的磁碟機是一個不存在的磁碟機。在下一個 DOS 指令中，指明正確的D引數。
- 槽號（Sn）引數所指定擴充槽並沒有插磁碟控制卡（disk controller card ）。現在錯誤的槽號成爲預設值，且 DOS 假設這個沒有連上此槽的磁碟片仍在使用中。即使下一個 DOS 指令指定正確的擴充槽， DOS 將永遠在等候這片不存在的磁碟，以便完成上個指令的動作，此情況已相當於系統“當掉”（ down ）了。如果記憶體中沒有您想要保存的程式，則只要重新啓動 DOS 即可。如果要使您的程式完整地復原，則必須依循以下的步驟操作：

1. 按下 RESET 鍵。
2. 打入 C A T A L O G SPACE S n，其中 n 是正確的槽號。

如果這時出現 SYNTAX ERROR ，表示 DOS 已被切斷。必須重新接通 DOS ，如果是處於 BASIC 提示狀態，請輸入

C A L L SPACE 1 0 0 2

如果是處於監督程式的星號狀態，請輸入

3 D 0 G（這是零不是字母 o ）

3. 然後再使用CATALOG指令——CATALOG，Sn——則一切都沒有問題。

無論如何，利用 VERIFY 指令確定您所使用的檔案已正確無誤存於磁碟片。如果儲存資料時發生錯誤但資料仍在記憶體中，請重新再儲存一次。

9 DISK FULL

DISK FULL 表示 DOS 要儲存資訊到磁碟時，發現沒有足夠的空間可用。在此種情況下， DOS 關閉（ close ）所有檔案，並保存所有可能的資訊。

當您得到 DISK FULL 訊息，您可以刪除磁碟上一、兩個檔案，然後再將記憶體中的資訊儲存於磁碟，或者更換磁碟片，再將記憶體中的資訊存入有足夠空間的磁碟片。

如果您接到此訊息之後，又再一次試圖將檔案存入已滿的磁碟片，則磁碟檔案目錄中一個已存在的檔案其磁區長度會設成零。除了目錄的某項目上會出現奇特的現象外，已存在的檔案不會受到任何損壞。

10　FILE LOCKED

FILE LOCKED　表示您要儲存（ save ）、寫入（ write ）、修改、增添（ append ）或刪除（ delete ）已鎖住的檔案（ locked file ）。檢查檔案目錄（ catalog ），是否在此檔案的檔名之前有星號標記（ * ）。要爲檔案解鎖，可利用UNLOCK指令。

11　SYNTAX ERROR

SYNTAX ERROR　表示您的程式使用 DOS 指令時，用了不正確的值或不正確的間隔字元（ separator ，如逗號或空白字元）。如果指令沒有必要的引數，也會出現此訊息。先檢查指令的語法對不對，然後再試一次。

產生這個錯誤訊息，最普通的原因是在指令中打錯字，只要檢查是否有拼錯，即可再試一次。

12　NO BUFFERS AVAILABLE

每次開啓（ open ）檔案以及每個 DOS 指令在執行時（ PR# 、IN#、MAXFILES 除外）都需要使用檔案保留區（ file buffer ）。NO BUFFERS AVAILABLE 表示您或您的程式要開啓的檔案太多，或下達的 DOS 指令需用的保留區已經超出記憶體中可用的保留區數量。

下達 CLOSE 指令，即可釋放檔案保留區。您可以利用MAXFILES 指令來增加檔案保留區的數量。要知道如何增加或減少檔案保留區的數量，請參考本書第 5 章有關 MAXFILES 指令。

13　FILE TYPE MISMATCH

FILE TYPE MISMATCH　表示 DOS 指令中所指定的檔案其檔

案型態（ file type）不適合此指令使用。

利用CATALOG指令檢查此檔案的型態，然後核對表B－2，以確保這個指令使用此種型態的檔案是合法的。

如果您確定指令是正確的，則使用不在此磁碟的某個檔案名稱，或使用不同的磁碟片，或將已存在檔案改名，或刪除已存在的檔案。

表C－4　各指令所用的檔案型態

指令	合法的檔案型態
LOAD, RUN, SAVE	A or I
CHAIN	I
OPEN, READ; WRITE, CLOSE APPEND, POSITION, EXEC	T
BLOAD, BRUN BSAVE	B

14 **PROGRAM TOO LARGE**

PROGRAM TOO LARGE 表示 DOS 要載入磁碟的檔案時，發現主記憶體沒有足夠的空間來容納此檔案的內容。

如果目前正處於“立即執行模式”，則試著下達 FP 或 INT 指令（依照您的程式型態選用其中一個適當的指令。）當 DOS 執行 FP 或 INT 時，會檢查記憶體，合併所有可用的空間，這樣做會將某些程式所造成的小片段消除。

註：FP 和 INT 指令是用來切換兩種不同的 BASIC 語言。請參考您的 BASIC 手冊。

利用MAXFILES指令減少檔案保留區的數量，使您的BASIC程式可用較多的記憶體。

如果您所執行的程式出問題，也會得到此訊息。也有可能上一個程式設定HIMEM: 或LOMEM: 的值，使您目前這個程式無法載入。您可以利用POKE，將目前正確值放入HIMEM: 或LOMEM: 的位址。有關的使用方法，請參考本書第6章。(註：HIMEM是您的程式可用的最高記憶體位址。有關您的系統HIMEM的最大值與DOS和3個檔案保留區記憶體使用情形，可參考第6章。)

順便一提： DOS要決定一個程式能否容下記憶體時，它只是檢查程式所佔用的磁碟磁區數。通常，程式不會將檔案的最後一個磁區填滿，但是DOS不管這些。DOS比較程式末尾對應位址的高位元組和LOMEM(Integer)或HIMEM(Applesoft)的高位元組，看看是否相同。意即程式本來可以容下記憶體，甚至還會留下少於256位元組的記憶體，但是載入時會出現此訊息。或許您可以在載入程式之前作某些修正，將HIMEM或LOMEM稍加移動，以更改其高位元組內含值。

15　NOT DIRECT COMMAND

NOT DIRECT COMMAND 表示您利用“立即執行”的方式下達APPEND、OPEN、POSITION、READ或WRITE指令。這些指令只能用於程式中的PRINT指述，請參考本書2、3、4章。

這個訊息會出現，也可能是程式停止執行又重新啓動。利用POKE指令可以修正此錯誤。請輸入

```
POKE 51; 0:CONT.
```

附錄D
程　式

D.1 SYSTEM MASTER磁碟內的各種程式

下列是 SYSTEM MASTER 磁碟的程式：

名　稱	功　能
HELLO	一 Applesoft 型態的歡迎詞程式（greeting program），祇要在 DOS 啓動後，而 Applesoft BASIC 語言已可用之情況下，即自動執行本程式。
APPLESOFT	一 Integer BASIC 型態的歡迎詞程式，祇要在 DOS 啓動後，而 Applesoft BASIC 語言已知無法使用之情況下，即自動執行本程式。
BOOT 13	一機器碼程式，用在載入"13磁區"舊型式的磁碟，該磁碟是利用較早期的 DOS 程式（DOS 3.2）產生的。
CHAIN	一機器碼程式，它可以載入並執行第二個程式，但不破壞（清除）已存於記憶體內第一個程式的變數及陣列的內含值。兩個要鏈接（chained）的程式其型態必須是 Applesoft BASIC。

CONVERT13	一Applesoft 型態之程式，它執行MUFFIN程式，將一 13 磁區的磁碟轉換成 16 磁區格式的磁碟。
COPY	在使用 Integer BASIC語言情況下，利用本程式來複製整片磁碟。
COPY.OBJ0	一機器語言的常式（routine），被COPY或COPYA程式所叫用。
COPYA	在使用Applesoft BASIC語言情況下，利用本程式來複製整片磁碟。
FID	一機器碼程式，合聚 DOS 提供的幾種指令功能的程式。
FILEM	一機器碼程式，它可啓動FID的執行，您可以一次同時操作數個檔案。
FPBASIC	存於磁碟片的Applesoft BASIC語言程式，其型態是機器碼。
INTBASIC	存於磁碟片的 Integer BASIC 語言程式，其型態是機器碼。
LOADER.OBJ0	存在 SYSTEM MASTER 磁碟上，經由HELLO載入記憶體的機器語言程式。LOADER.OBJ0的作用是將另一套 BASIC 語言載入語言界面卡或者記憶體內。假若已存在著第二套語言系統，則不發生效用。LOADER.OBJ0同時將存於 ROM 的監督程式，轉存至語言界面卡或記憶體的 \$F800 位址上。
MASTER	一個啓動MASTER CREATE程式（其型態是機器碼）的 Applesoft BASIC程式，將一已啓始化的磁碟片轉變成“主”磁碟（master disk）。
MASTER CREATE	一機器碼程式，可產生一主磁碟，亦即是內含有可將 DOS 本身程式儘量放在記憶體中較高位置的 DOS 系統磁碟。一主磁碟可適用於任何記憶體容量的Apple II電腦。

MUFFIN	可轉換"13 磁區"磁碟至"16 磁區"格式的磁碟之機器碼程式。
RENUMBER	一Applesoft 程式，可將一 BASIC 程式或合併的兩個程式其列號（line number）重新加以編號。
SLOT #	一Applesoft 程式，可回報目前的擴充槽及磁碟機預設值。
START13	一Applesoft 程式，可啓動BOOT 13程式之程式，其目的是啓動先前版本DOS 產生的13磁區磁碟。

D.2 SAMPLE PROGTRAMS磁碟內的各種程式

下列是SAMPLE PROGRAMS磁碟的程式：

名 稱	功 能
ADDRESS	用Applesoft BASIC 撰寫的示範用程式，用以闡示"隨意存取檔案"的讀取和寫入動作。
ANIMALS	用 Integer BASIC 撰寫的遊戲程式，可用來建立動物名稱的資料檔案。
APPLEPROMS	RANDOM程式須用到的資料檔案，它含有有關於物料庫存的資料錄。
APPLEVISION	用 Integer BASIC撰寫的示範用程式。
BLACK.BOOK	儲存ADDRESS程式中須用到的資料錄之資料檔案。
BRICKOUT	以Applesoft BASIC 撰寫的遊戲程式。請利用鍵盤的方向鍵操作本遊戲，或搖桿來控制方向。
COLOR TEST	以Applesoft BASIC 撰寫的示範用程式，可藉

	它調整彩色電視的螢幕特性。
DELETE.ME.1	Applesoft 示範用程式，讓您練習刪除一個檔案。
DELETE.ME.2	Applesoft 示範用程式，讓您練習刪除一個檔案。
DELETE.ME.3	Applesoft 示範用程式，讓您練習刪除一個檔案。
EXEC DEMO	Applesoft 示範用程式，告訴您一個EXEC程式檔案是如何產生的，及如何執行它。
FPBASIC	存於磁碟的 Applesoft BASIC 語言程式，機器碼檔案。（譯註：FPBASIC 即是習稱的浮點式 BASIC 直譯程式，Interpreter ）
GET TEXT	Applesoft 示範用程式，可讀取一文字檔的內容。
HELLO	以 Applesoft 撰寫的歡迎詞程式。
INTBASIC	存於磁碟的 Integer BASIC 語言程式，機器碼檔案。
LOADER OBJ0	在 SYSTEM MASTER 磁碟上，經由HELLO 載入記憶體的機器語言程式。LOADER. OBJ0 的作用是將另一套 BASIC 語言載入語言界面卡或者記憶體內。假若已存在著第二套語言系統，則不發生效用。LOADER. OBJ0 同時將存於 ROM 的監督程式，轉存至語言界面卡或記憶體的 $F800 位址起的區域。
LOCK.ME.1	Applesoft 示範用程式，讓您練習鎖定一尚未鎖住的檔案。
LOCKED.UP.1	Applesoft 示範用程式，讓您練習解除一已鎖定檔案的鎖住保護。
LOCKED.UP.2	Applesoft 示範用程式，讓您練習解除一已鎖定檔案的鎖住保護。
MAKE TEXT	Applesoft 示範用程式，闡示一循序文字檔的建立過程。
ONERRDEMO	展示“錯誤復原”（ error recovery ） 技巧的樣本程式。它檢查某一檔案是否已設定鎖住保護，若是的話，則解除其鎖定狀態。

POKER	Applesoft 示範用程式，闡示機器語言程式轉換成BASIC程式的處理過程。
PHONE LIST	Applesoft 示範用程式。
RANDOM	Applesoft 示範用程式，闡示隨意處理文字檔的讀取和寫入動作。它利用APPLE PROMS記錄使用到的資料。
VERIFY.ME	Applesoft 示範用程式，讓您練習查證一檔案是否已正確地存入磁碟內。

附錄E

DOS操作概念及指令摘要

本附錄中有關於“DOS使用者手册”(DOS User's Manual)的部份以“第X章，DU”標示之，其他在本書(“程式設計者手册”)中介紹的各種指令，以“第X章”標明在本書中出現的章節。

E.1 操作概念

DOS可在Apple II 電腦的所有機種上運轉，包括Apple II，Apple II Plus 以及 Apple IIe。若有特別適用於某種機型的資訊流程和操作程序，將會在適當的章節中明示。

E.1.1 冷啓動

“啓動”是將您的電腦扭開電源開關的操作過程(或者模擬的相當過程)，以便讓DOS載入記憶體內，其步驟如下：

1. 將DOS磁碟片插入第一部磁碟機(Drive 1)。
2. 開啓顯示裝置的開關，如電視機或顯示器(monitor)。

3. 找到電腦的電源開關，並打開它。

對於 Apple IIe 機型而言，在電源已開啓的情況，可以進行上述“冷啓動”（cold start）的模擬開機過程，其程序是按住[Apple]，再按下[CONTROL]-[RESET]，放掉[RESET]，[CONTROL]，最後鬆開[Apple]。

對於標準的 Apple II 或者 Apple II Plus 機型的冷啓動模擬過程，其程序是打入Sn（若是擴充槽6號，即打入6），[CONTROL]-[P]，[CONTROL]-[L]，或者Cn00G。其中的n即代表實際上插有磁碟機控制界面卡的擴充槽編號。

參考“第一章，DU”。

E.1.2 溫啓動(所有 Apple II 機種)

當電源已通的情況下，您可以重新啓動電腦，其方法如下：同時按住[CONTROL]和[RESET]鍵，而後再放開它們。在較早期的 Apple II 機種，您祇需按下[RESET]便可。

電源已開啓的再啓動電腦之過程，常稱之為“溫啓動”（Warm start），並不會再度載入 DOS 至記憶體內，通常也不會清除已存在於記憶體的程式。（譯註：溫啓動與冷啓動之差異點在目前電腦電源“已開啓”或“未開啓”）。

E.1.3 啓始化一磁碟片

磁碟片啓始化的過程是：

1. 啓動 DOS，並將一空白磁碟片放入磁碟機內。
2. 下達 BASIC 的 NEW 指令，以清除記憶體的任何內容，

而後，再打入一歡迎詞程式，例如

```
10 PRINT "32K DISK INITIALIZED 5 MAY 84"
20 END
```

請打入RUN，測試該程式是否鍵入無誤。

3. 假設您想命名歡迎詞程式爲HELLO，則打入INIT HELLO. 它可以規劃一磁碟的格式並將該程式以HELLO爲檔名儲存至磁碟上。

參考“第一章，DU”。

E.1.4 產生一主磁碟

下列步驟是轉換一已啓始化磁碟爲主磁碟的過程：

1. 首先設計好一歡迎詞程式（稍後，您可以感覺到“它變了”，即是產生了一主磁碟。您可以修改先前的歡迎詞再以新檔名存之（例如 SAVE BIG HELLO ）或者設計一新的程式。自磁碟機移走磁碟。
2. 將 SYSTEM MASTER 磁碟放入磁碟機，打入 RUN MASTER 。
3. 請打入您的歡迎詞程式的新檔名，即是步驟1中SAVE指令用到的名稱。假若您希望磁碟片是可以啓動其他種類之程式——亦即是產生一“轉鑰式磁碟（ turnkey ）”—請按下 ESC 而後再鍵入該程式的名稱。
4. 取走SYSTEM MASTER磁碟並代以您希望的已啓始化磁碟，按下 RETURN 鍵即開始進行轉換動作。

請參考“第5章”。

E.1.5 解譯DOS指令

“延遲執行”(deferred execution)模式的指令是指在 BASIC 程式中下達指令的狀態，相對的“立即執行”(immediate execntion)模式是指從鍵盤下述指令的狀態。假若一指令以 CONTROL - D 前導的話，由 DOS 負責來解說它，下達延遲執行式的指令之方法，是利用 PRINT 指述來印出由 CONTROL - D 和後隨的指令名稱所組成的字串，如下例所示：

```
10 D$=CHR$(4)
20 PRINT D$;"CATALOG"
```

```
10 PRINT "CATALOG"
20 REM (There's an invisible
30 REM CONTROL-D before
40 REM the C inside first quote
```

在 BASIC 中的CHR$(4)函數即是相當於 CONTROL - D 字元的效果。

請參考“第2章”。

E.1.6 磁碟容量

DOS 使用者最多可用到496磁區以儲存資訊。每一磁區最多可儲存256位元組的資訊。

一空的文字檔(empty text file)是佔用1個磁區——內含有該檔案的磁軌／磁區的使用表(track/ sector list)。空的Applesoft, Integer BASIC 和機器碼檔案(binary file)則是佔用2個磁區——其中一個內含磁軌／磁區的記錄項，另一

個則記載程式之長度，它是此程式所在的第一個磁區。

請參考"附錄B"。

E.1.7 檔案型態

檔案型態（file type）是在磁碟的目錄部份以代碼（A、I或T之類）加以標示。

E.2 指令記號之說明

本章節簡介在本書中使用的各種記號及其代表意義。

E.2.1 記號意義

UPPERCASE	大寫字體，表示是一指令的實際名稱，請完全忠實地依字面打入之。
lowercase	小寫字體，表示在指令中須由您打入的任何適當名稱，例如一程式之名稱。
[ ]	括弧內項目，代表它是指令中可有可無的選擇項（option），您可依據情況，視需要選用它或者略去不用。不管如何，請不要打入括弧本身。

E.2.2 語　法

"檔案名稱"通常是一指令名稱之後的第一個引數，其餘的引數其排列次序並無關緊要。檔案名稱之後須加上逗號以分離其他後隨的引數，如下列所示：

INIT fn[,Vn][,Sn][,Dn] 代表指令的語法。

INIT HELLO,V17 代表該指令的典型例子。

- 指令名稱 INIT 是大寫字體，故完全依字完整地鍵入之。
- fn 項引數代表"file name"（檔案名稱），以實際的某檔案名稱取代語法中小寫字體的部份，本例是 HELLO。
- Vn 項引數是可有可無的卷號選擇項。在本例中，Vn部份以 V17 取代之。（此卷號是任意選取的。）
- Sn 項引數是可有可無的擴充槽編號選擇項，本例中並無此項。
- Dn 項引數，亦是一選擇項，代表DOS存取資訊所在的磁碟機代號，本例是第二部磁碟機（drive 2）。

請參考"第2章"。

E.2.3 引 數

fn 出現在指令語法中，表示須由您打入的檔案名稱。一檔名必須以英文字母開頭，可含有 1 到 30 個字元。除了逗號以外的任何可顯示（印出）的文字均可以是檔名的一部份。

n 若當作PR#和IN#指令的引數，則表示數目 1 到 7 的一個擴充槽編號。若當作MAXFILES指令的引數，則代表 1 到 16 之間的可用保留區（buffer）數目。您可以打入平常的數值或者十六進制的數值。

n 在其他引數出現時，它可以是十進制或十六進制數目。十六進制數目是以錢號（$）前導的，例如$FE。

（以上三項，參考"第2章"。）

An　代表一記憶體位址（ Address ）。請以 0 到 65535之間的數值取代此記號。與BSAVE 指令合用時，表示它是開始轉存至記憶體的位址。在BLOAD和 RUN 指令中，則表示載入機器碼檔案至記憶體的目的位址。若略去本項，則 DOS 負責將程式放在曾以BSAVE 指令指明的位址爲存放之起始位址。

請參考“第 6 章”。

Bn　代表一位元組（byte）或字元（character ）的數值。請以 0到 32767之間的數值取代此記號。若略去本項，則 DOS 預設爲 0 。在循序檔案的存取中，Bn 是指明在檔案中的某位元組的絕對位址；在大多數程式裡。它的值不會超過檔案的最後一個位元組，在隨意存取檔案裡，表示由Rn 所指明的某一資料錄（ rewrd）內的某位元組絕對位址；在大多數程式裡，其值不會大於目前資料錄的最後一個位元組。

（循序存取檔案的說明，請參考第 3 章，隨意存取檔案則參考第4 章。）

Dn　代表一磁碟機（ Drive ）的代號。請以 1 或 2 取代此代號。磁碟機代號起先是預定爲1 ，而後即設定爲最近剛用到的 DOS 指令中Dn項引數值。

請參考“第 2 章”。

Ln　代表長度（ length ）。一位元組或字元的數目，其值是0到 32767之間的數值。它用在開啓一隨意存取的文字檔（OPEN 指令），表明每一資料錄的固定長度（單位是位元組，Byte ）。在BSAVE 指令中，是表明以An 起始的記憶體位址，共有Ln個位元組。若略去Ln 項引數，則 DOS 預設爲 1 。

請參考“第 6 章”。

Rn　用在 POSITION 和EXEC指令中，表示需跳過多少資料欄

(field) 或者多少列的指述 (相對於目前位置)。以 0 到 32767之間的數目代替 n 。若略去本項， DOS 預設成 0 ，表示一檔案的起始端。

(POSITION 指令見第 3 章，EXEC 指令見第 5 章。)

Rn　在隨意存取檔案的READ和WRITE指令中，代表一資料錄號碼。以 0 到 32767 的數值取代 n 。在 OPEN指令之後，Rn 起始預設爲 0 ，表示一檔案的第一個資料錄，而後，它即設定成存取後上一個資料錄的編號。Rn 是代表一隨意存取檔案的資料錄 (絕對) 號碼。

(隨意存取檔案的部份參考第 4 章。)

Sn　代表磁碟機所在的擴充槽編號 (Slot) (插有磁碟機界面控制卡)。以 1 到 7 之數代入 n 。若略去本項， DOS 使用預設的擴充槽編號。 DOS 起始設定啓動時所在的磁碟機之擴充槽編號爲預設值，而後，即設成最近一個指令的 Sn 項引數值。

請參考第 2 章。

Vn　代表磁碟片的卷號 (Volume)。以 1 到 254 之數代入 n 。DOS 起始設定磁碟的卷號爲啓動時該磁碟的卷號，而後，即設成最近一個指令的 Vn 項引數或讀取之磁碟卷號。當在 INIT 指令略去本項時， DOS 預設其卷號爲 254 ，至於其他 DOS 指令略去本項，或者僅指明卷號爲 0 ，或者打入V一字，則 DOS 一概不理不正確的卷號說明規格。

請參考第 2 章。

E.3　指令摘要

在本節中，DOS 指令依據性質分成下列 5 類：

1. 存取磁碟內容的非程式化指令（ nonprogramming command ）：

CATALOG	INIT	DELETE	RENAME
LOCK	UNLOCK	VERIFY	RUN
LOAD	SAVE	PR#	IN#

2. 可控制程式所在環境的指令：

FP	INT	MON	NOMON
EXEC	MAXFILES	CHAIN	

3. 循序存取文字檔的程式指令：

OPEN	CLOSE	READ	WRITE
APPEND	POSITION		

4. 隨意存取文字檔的程式指令：

OPEN	CLOSE	READ	WRITE

5. 機器碼檔案的指令：

BRUN	BLOAD	BSAVE

下列是所有 DOS 指令的簡略解說，每一指令均加註一個例子，同時在解說之後頭附上可找到更詳細說明的章節號碼。（譯註：可程式化指令與非程式化指令之差異點是在 BASIC 的PRINT 指

述中能否用CONTROL—D(亦即是CHR$(4))方式呼叫該指令)。

E.3.1 存取磁碟內容的非程式化指令

CATALOG [,Sn] [,Dn]

例：CATALOG

在螢幕上顯示磁碟片的卷號以及所有存於磁碟的檔案名稱，並包括每一檔案的型態，佔用的磁區大小、該檔案是否已鎖定保護(星號＊加註在檔案型態之旁表示“已鎖定”)。

檔案型態分別是：

I　Integer BASIC 程式檔案，以SAVE指令產生的。
A　Applesoft BASIC 程式檔案，以SAVE指令產生的。
T　文字檔(text file)，以OPEN指令配合WRITE指令產生的。
B　二進碼記憶體映像檔(binary memory-image file)，以BSAVE指令產生的。(譯註：本書譯名取“機器碼檔案”)
R　可重定位的(relocatable)組合語言程式檔，以BSAVE指令產生的。
S　保留，爲以後之用。

若某檔案的長度超過255磁區，則顯示的目錄其檔案長度部份是超過255後自零起算的部份，它可能會給予您錯誤的長度印象，因而錯估了剩餘的磁碟空間。

請參考“第3章，DU”。

INIT fn [,Sn] [,Dn] [,Vn]
例：INIT HELLO, V18

將磁碟的表面規劃成磁軌和磁區，並將所有資料欄的內容設定成零。 INIT 同時也將 DOS 本身的程式及歡迎詞程式存入磁碟內。

請參考" 第 3 章，DU "。

DELETE fn [,Sn] [,Dn] [,Vn]

例 ：DELETE TEXT

自磁碟中刪除一未鎖定的檔案。假若指名的檔案不存在於磁碟中， DOS 顯示 FILE NOT FUUND 錯誤訊息，欲避免此項錯誤，在DELETE指令之前下達OPEN指令。

請參考" 第 4 章，DU "。

RENAME fn1,fn2 [,Sn] [,Dn] [,Vn]

例 ：RENAME CURRENT, ARCHIVAL, S7, D1

更名一檔案的檔名 fn1 (CURRENT) 爲新檔名 fn2 (ARCHIVAL)，但不會影響到檔案的內容。假若此檔案已開啓，則自動將它關閉，而後進行更名動作。RENAME 並不負責先行查驗 fn2 的檔案是否已存在於磁碟上。

請參考" 第 4 章，DU "。

LOCK fn [,Sn] [,Dn] [,Vn]

例 ： LOCK LOVE LETTERS, V31

設定鎖住保護以預防無意中破壞檔案。已鎖定的檔案在磁碟目錄上加註有星號 (*) 您無法更動檔案名稱、刪除或變動其內含資訊。

請參考" 第 4 章，DU "。

UNLOCK fn [,Sn] [,Dn] [,Vn]

例 ：UNLOCK DOORS, D2

解除一檔案的鎖定保護，以便往後能更動檔名，刪除它或者變動其內含資訊。

請參考“第 4 章，DU ”

VERIFY fn [,Sn] [,Dn] [,Vn]

例：VERIFY FACTS

測試 DOS 是否能自磁碟中讀取檔案並存入記憶體內。fn 必須是未設定鎖住保護。當 DOS 能夠讀取檔案，DOS 並未顯示任何“確認無誤”的訊息，反之，則顯示 I/O ERROR 錯誤訊息。您可以查證任何型態的檔案。

請參考“第 4 章，DU ”。

RUN fn [,Sn] [,Dn] [,Vn]

例：RUN ANNUITY, D1

將一 BASIC 程式（其型態是 A 或 I ）自磁碟拷貝出來存入記憶體內，並立即執行它。

請參考“第 5 章，DU ”。

LOAD fn [,Sn] [,Dn] [,Vn]

例：LOAD CARGO, S6, D1

將一程式（其型態是 A 或 I ）自磁碟拷貝出來並存入記憶體內。一旦程式已存於記憶體內，您可以執行它，修改它，或者再儲存爲磁碟檔案。

在載入新程式至記憶體之前，DOS 先關閉目前所有已開啓的文字檔。當一新程式眞正地載入記憶體之時，DOS 已清除目前既存於記憶體的任何內容並切換成該程式之型態所屬的 BASIC 語言。

假若程式之檔案型態爲 A，而 Applesoft 並不存於記憶體或者無法自 Applesoft 韌體（ firmware ）ROM 界面卡取得，

DOS 立即設法自磁碟中載入該語言並執行它。如果 Applesoft 語言也不存於磁碟，則顯示 LANGUAGE NOT AVAILABLE 訊息

請參考“第 5 章，DU ”。

SAVE fn [,Sn] [,Dn] [,Vn]

例：SAVE COLOR DEMOS, D2

將目前存於記憶體的 BASIC 程式寫入磁碟成為一檔案。當該程式是Applesoft 程式，DOS 在儲存它時設定檔案型態為 A，同理，若是 Integer BASIC 程式，則以檔案型態 I 儲存之。

當指名的 fn 檔案尚未存在於磁碟，DOS 負責產生您指定檔名的檔案，並將目前既存於記憶體的內容存入該檔案。倘若磁碟已有一相同檔名且相同型態的檔案，則 DOS 將目前 BASIC 程式覆蓋寫入此檔案，如果檔名相同個址態相異，則顯示 FILE TYPE MISMATCH 錯誤訊息。

請參考“第 5 章，DU ”。

PR# n

例：PR# 6

將原先送至螢幕顯示的字元改道傳送至連接於擴充槽 n 號的裝置。# 記號是指令名稱的一部份，您必須打入此字元。

在標準的 Apple II 和 Apple II Plus 機種，PR #0指令回復輸出至螢幕的正常情況在 Apple IIe機種，若不使用 80 格文字顯示卡，PR #0 是回復 40 格螢幕的情況，但假若 80 格文字顯示 卡希望發生效用，則請用PR #3指令回復80 格螢幕的情況，或者 PR #3之後按下 ESC CONTROL–Q 回至 40 格顯示畫面。

當延遲執行模式的PR#指令並未以 CONTROL–D 前導，或者在編號 n 的擴充槽裡並無某裝置的界面控制卡，則 DOS 無法與之連結，您只好重新啓動 DOS 。

請參考“第 2 章”。

IN# n

例 ：IN# 2

自連接於編號 n 的裝置（例如，一外接的終端機）讀取字元，換句話說亦即是自“鍵盤”讀取字元的資訊管道改變成自“ n 號擴充槽的裝置”讀取之。# 記號是指令名稱的一部份，您必須打入此字元。

若想回復至鍵盤的正常讀取管道，則下達IN #0指令。當延遲執行模式的PR #指令並未以 CONTROL—D 前導／或者在編號 n 的擴充槽裡並無某裝置的界面控制卡，則 DOS 無法與之連結，您只好重新啓動 DOS 。

請參考“第 2 章”。

E.3.2 控制程式環境的指令

FP [,Sn] [,Dn]

例 ：FP, D2

切換至Applesoft語言系統。它會重新設定存於記憶體內指引先前 BASIC 程式和各變數之指數之指標值；亦即是記憶體內該程式相當於漏失了。

若您的 Apple II 電腦含有Applesoft韌體界面卡或語言界面卡，DOS 就可在界面卡找到Applesoft。假若您的電腦並未上述界面卡，則 DOS 嘗試自磁碟讀取 FPBASIC 程式並執行它。

請參考“第 2 章”。

INT

例 ：INT

切換至 Integer BASIC 語言系統，它會重新設定存於記憶體內指引先前 BASIC 程式和各變數之指標值；亦即是該程式相當於漏失了。若並無 Integer BASIC 語言系統可用，則您得到的是 LANGUAGE NOT AVAILABLE 錯誤訊息。

MON [C] [,I] [,O]

例：MON O and MONICO

顯示電腦與磁碟之間溝通的動作訊息。C 項表示顯示其間的磁碟指令（Command），I 項則顯示自磁碟讀取並傳送電腦的輸入（Input），O 項則顯示送至磁碟的輸出（Output）。您可以選用上述三選擇項引數，以任何的排列次序及組合，但至少須有其中一項；用來分隔引數的逗號則可有可無。

MON 指令選用的引數之效果一直持續至您下達了NOMON，FP 或 INT 指令，或者您重新啓動系統，執行某一程式並不會取消它們的作用。

請參考"第 2 章"。

NOMON [C] [,I] [,O]

例：NOMON C and NOMON I, C

抑制（取消）先前由 MON 指令選用的引數效果，不再顯示電腦與磁碟之間溝通的動作訊息。C 項表示抑制磁碟指令的顯示，I 項則抑制自磁碟讀取並傳送電腦的輸入顯示，O 項則抑制送至磁碟的輸出顯示。您可以選用上述三選擇項引數，以任何排列次序及組合，但至少須有某一項，用來分隔引數的逗號則可有可無。

NOMON C, I, O指令即相當於系統回復至起始狀態：在磁碟與電腦之間的指令及資訊輸出入流程不顯示於螢幕上。

請參考"第 2 章"。

EXEC fn [,Rn] [,Sn] [,Dn] [,Vn]

例 ：EXEC UTILITY

指示 DOS 自一 BASIC 程式產生的指令檔（command file）取得指令列（line）加以執行。（正常情況下，您是從鍵盤打入指令列。）fn 即是指令檔的檔名，其內容可以是 DOS 指令，資料和 BASIC 指述。

DOS 自指定的第Rn列（line）之處開始執行此指令檔。假若您略去Rn項引數。DOS 自檔案的第一列（相當於Record 0）開始取得指令敍述。假若R字之後的資料錄編號大於檔案的結尾資料錄，則顯示 END OF DATA 錯誤訊息。

當一程式是在指令檔的控制之下執行時，任何程式中的INPUT指述將是自指令檔中讀取下一資料欄內的東西，而不是自鍵盤取得輸入值，假若這個輸入是可立即執行的 DOS 指令，則在繼續此程式之前執行完這個指令。

請參考“第5章”。

MAXFILES n

例 ：MAXFILES 6

n是1到16之間的整數，表示同一時間內同時有作用（開啓）的檔案個數。當MAXFILES執行過後，DOS 即爲每個檔案預留一檔案保留區（595位元組的記憶體區域）。當 DOS 啓動時，系統自動預留3個檔案保留區（1785位元組），亦即是您最多可同時使用3個檔案。

註：使用MAXFILES會移動HIMEM的位置，清除記憶體中的Integer BASIC 程式和Applesoft 的字串區域。請在載入和執行一程式之前，或者程式的第一列指述，使用MAXFILES 指令。

除了PR#，IN#和MAXFILES之外的所有 DOS 指令，在執行時需要1個檔案保留區(file buffer)。假若您指定 MAX-FILES 1則只能開啓一個檔案，大多數的 DOS 指令一執行即 會顯示 NO BUFFERS AVAILABLE 的錯誤訊息。

請參考“第5章”。

CHAIN fn [,Sn] [,Dn] [,Vn]

例：CHAIN PART TWO, D1, S7

執行一新程式但不關閉目前已開啓的檔案，它會保留先前程式所使用的變數內含值，使得下一個新程式能運用先前程式的結果再加以運算，得到新值再留給以後的程式。惟僅限 Integer BASIC 程式可用本指令與其他程式鏈結(chain)在一起。

E.3.3 循序存取文字檔的程式化指令

一文字檔(text file)是以逗號分隔的一連串資料欄所組合，儲存於磁碟上的檔案，其型態是T。(註：資料欄(field)是由1個到32767個字元構成的，其結尾是個“歸位字元”(Carriage return)。它可以有兩種存取的方式：循序存取(sequential access)或隨意存取(random access)。用WRITE指令來儲存資訊，用READ指令來擷取資訊。READ和WRITE指令須配合OPEN和CLOSE指令使用之。

在循序存取式的文字檔案裡，開啓一檔案並不需要說明資料錄的長度，同時每一資料欄是緊跟著上一資料欄存放的，亦即是DOS將每一資料欄的第一個字元緊跟著寫在上一資料欄的結束字元(

註：所有相關於循序存取的指令僅能用在“延遲執行”模式。僅有 CLOSE指令可同時適用“立即執行”和“延遲執行”模式。

歸位字元）之後。每一次此檔案開啓後， DOS 自檔案到起始點開始，循序地存取（讀取或寫入）各資料欄的內容。

OPEN fn [,Sn] [,Dn] [,Vn]

例 ：OPEN SESAME, D2

開啓一循序存取文字檔，以便 DOS 往後能讀取或寫入此檔案。當指名的檔案尚未存在於磁碟上， DOS 即產生一型態爲 T 的文字檔。

當您開啓一檔案時， DOS 即爲此檔預留一個檔案保留區（595 位元組的記憶體），並且設定讀取和寫入資料的位置指標爲檔案的起始端。

請參考“第 3 章”。

CLOSE [fn]

例 ：CLOSE CLOSET

告訴 DOS 您已經全部完成檔案的存取動作。在結束之前，程式必須關閉所有已開啓的檔案，以保證所有字元之已寫入檔案，並且收回檔案保留區以便他用。

假若程式有錯並在有能力關閉所有已開啓檔案之前就中止的話您只得從鍵盤下達 CLOSE 指令以關閉檔案。

請參考“第 3 章”。

READ fn [,Bn]

例 ：READ TLEAVES

告訴 DOS 讀取那一個循序存取文字檔，以及自那個位置開始讀取之。僅能在OPEN下達後才可使用READ指令，指名的 fn 檔案一直被讀取內含資訊，直到下一個 DOS 指令使用了不同的檔案名稱。

B項引數是告訴 DOS 自指定的位元組位置（Bn）開始讀取資訊，注意，是相對於目前位置的第Bn位元組。

順便提醒您，一個 INPUT 指述僅能一次讀取檔案的一個資料欄內容。曾提到一資料欄可內含1到32767個字元並以歸位字元結尾，然而 BASIC 對字串長度及輸出入的緩衝區長度限制，在讀取一超過255字元的資料欄時，可能會出差錯，請讀者特別留意此情況。

請參考“第3章”。

WRITE fn [,Bn]

例：WRITE ADDRESS.DATA

告訴 DOS 寫入那一個循序存取文字檔，以及自那個位置開始寫入之。僅能在OPEN下達後才可使用WRITE 指令；指名的fn檔案一直接受寫入的資訊，直到下一個 DOS 指令使用了不同的檔案名稱。

B項引數是告訴 DOS 自指定的位元組位置（Bn）開始寫入資訊，注意，是相對於目前位置的第Bn位元組。

順便提醒您，本指令下達後，由接著的 PRINT 指述負責輸出資訊至指定的檔案上。“輸出”的內容可能包含有“問號提示字元”，假若您在WRITE指令之後使用了 INPUT 指述（問號即是 INPUT 指述顯示的指示字元）；或者含有一串錯誤訊息之字串，假若您的程式裡並無錯誤情況處理的指述（ONERR GOTO)的話。

APPEND fn [,Sn] [,Dn] [,Vn]

例：APPEND INFO

開啓一循序存取文字檔以便您的程式自檔案的結尾處開始寫入資料字串。APPEND 指令開啓一可寫入的檔案，並設定“目前位址”爲檔案的結尾處。

本指令下達後，寫入的下一個字元會緊跟著先前已含有資訊的檔案最後一個字元。APPEND指令之後，須後隨一個沿用相同檔名引數的WRITE指令。

POSITION fn [,Rn]

例：POSITION ADDRESS.DATA, R277

設定讀取或寫入的“目前位址”。POSITION 指令的用意是讓您在準備讀取或寫入某資料欄之前，先向前跳過指定的資料欄數。

當R項引數略去不用時，“目前位址”即是檔案的起始處。若指定3 R項數值n，新的“目前位址”即是相當於現在的“目前位置”往前n個資料欄。

POSITION指令會掃瞄檔案的內容，一個一個字元的檢查，直到找到第n個歸位字元，將“目前位置”設定爲緊跟該歸位字元後的第一個字元，以後的READ和WRITE指令即自此位置開始讀取或寫入資訊。假若您嘗試將“目前位置”越過檔案的結尾處，則會出現 END OF DATA 錯誤訊息。

E.3.4 隨意存取文字檔的程式化指令

當開啓一隨意存取文字檔案，須選用“長度”引數，它指明每一資料錄（record）內所含的字元個數。

DOS 可以自任何指定的位置開始讀取檔案內容或寫入資訊，亦即是您不需先讀取所有在該指定位置之前的資料錄，也不必先讀取在指定的資料錄內該位置之先的各字元。DOS 可以不必顧到先前是在那一個資料錄作過存取動作，直接就可以存取任一個資料錄或資料欄。

警告： 所有適用於隨意存取文字檔的指令均是屬於“延遲執行”模

式。只有CLOSE指令是同時適用於立即執行和延遲執行兩種模式。

OPEN fn, Ln [,Sn] [,Dn] [,Vn]

例：OPEN SESAME, L2

開啓一隨意存取文字檔。假若指名的檔案不存在於磁碟上，DOS 就產生一型態T的文字檔。假若已存在，則它必須尚未開啓。

說明每一資料錄長度的Ln項引數是必要的引數。每一次開啓同一個檔案，須使用同長度的引數，DOS 即利用該值計算每一資料錄的起始位置。

當您開啓一檔案，DOS 立即為該檔案預留一檔案保留區（595位元組的記憶體區域），設定目前位置為檔案的起始點，並設定資料錄的字元個數為長度引數之值。

CLOSE [fn]

例：CLOSE ENCOUNTERS

告訴 DOS 您已經全部完成檔案的存取動作。在結束之前，程式必須關閉所有已開啓的檔案，以保證所有字元之寫入檔案，並且收回檔案保留區以便他用。

當您的程式有寫入檔案的動作，CLOSE 指令先將留在檔案緩衝區的所有字元寫入檔案內，而後才收回該保留區。當CLOSE指令並未指名某個檔案，DOS 則立即關閉除指令檔案之外的所有檔案。

假若程式有錯並在有能力關閉所有已開啓檔案之前就中止的話，您只得從鍵盤下達CLOSE指令以關閉檔案。

請參考“第4章”。

READ fn [,Rn] [,Bn]

例：READ PALMS,R3,B30

告訴 DOS 讀取那一個隨意存取文字檔，以及自那個位置開始讀取之。僅能在該檔案已開啓後才可使用READ指令；指名的 fn 檔案一直被讀取資訊，直到下一個 DOS 指令使用了不同的檔案名稱。

R項（表示資料錄編號，Record number）引數是指定檔案開始讀取的資料錄編號。若略去R項，DOS 預設其值爲 0 ，亦即是第一個資料錄。

B項（ Byte ）引數是指定在R項引數指明的資料錄內第幾個位元組之位置。若略去B項， DOS 預設其值爲 0 ，亦即是資料錄內第一個位元組。

順便提醒您，本指令下達後，由Applesoft的INPUT 指述或 GET 指述負責從檔案（不是鍵盤）讀取資訊。INPUT 指述是自目前資料錄讀取一個資料欄的內含字元，一次一個資料欄。

請參考“第 4 章”。

WRITE fn [,Rn] [,Bn]

例：WRITE ADDRESS.DATA, R3

告訴 DOS 寫入那一個隨意存取文字檔，以及自那個位置開始寫入之，僅能在檔案已開啓後才可使用WRITE指令。指名的 fn 檔案一直被寫入資訊，直到下一個 DOS 指令使用了不同的檔案名稱。

R項（表示資料錄編號）引數是指定檔案開始寫入的資料錄編號。若略去R項，DOS 預設其值爲 0 ，亦即是第一個資料錄。

B項（Byte)引數是指定在R項引數指明的資料錄內第幾個位元組之位置。若略去B項， DOS 預設其值爲 0 ，亦即是資料錄內第一個位元組。

順便提醒您，本指令下達後，原先應送到螢幕顯示的任何字元均送到（寫入）檔案內；其輸出內容可能包含有“問號指示字元”

，假若您在WRITE指令之後用了INPUT指令的話（問號即IN-PUT指述顯示的提示字元），或者含有一串錯誤訊息之字串，假若程式中並無錯誤情況處理的指述（ONERR GOTO）的話。

請參考“第4章”。

E.3.5　二進制機器碼檔案的指令

BRUN fn [,An] [,Sn] [,Dn] [,Vn]

例：BRUN SUPER, A$C0A, V75

將一二進制機器碼檔案（其型態為B）自磁碟中轉存至記憶體任一指定的區域，並立即執行它。

當選用了A項引數，即表示自指定的記憶體位址（Address）開始儲存資料。若省略A項引數，則DOS即自先前儲存此檔案（BSAVE指令）的起始位置，開始將資料載入記憶體內。

請參考“第6章”。

BLOAD fn [,An] [,Sn] [,Dn] [,Vn]

例：

BLO PICTURE, A8192	（十進制表示法）
BLOAD PICTURE, A$2000	（十六進制表示法）

將一二進制機器碼資料自磁碟中轉存至記憶體任一指定的區域。

當選用了A項引數，即表示自指定的記憶體位址（Address）開始儲存資料。若省略A項引數，則DOS即自先前儲存此檔案(BSAVE指令）的起始位置，開始將資料載入記憶體內。

注意，假若您選用了A項引數，一機器語言的程式載入記憶體後，可能無法自指定的新位址順利地被執行。

BSAVE fn , An , Ln [,Sn] [,Dn] [,Vn]

例： BSAVE PICTURE, A16384, L8192　（十進制表示法）
BSAVE PICTURE, A$4000, L$2000　（十六進制表示法）

將一存於記憶體的機器碼資料存入磁碟內。當指名的檔案不存在於磁碟，**DOS** 即產生一機器碼檔案（B 型態）；假若已存在，則指定的記憶硬部份將會覆蓋寫入原檔案的內容。An（表起始位址，n 即是位址值）與 Ln（表長度，n 即是位元組個數）引數可以讓您指定記憶體的任一部份。

請參考“第 6 章”。

附錄 名詞解釋

本附錄包括出現在本書中的術語與其他會出現在Apple 相關書籍的名詞。本書中，每當一個重要的名詞第一次用到時，都以粗體字出現，表示這個名詞將在此附錄中闡釋其意義。

address （位址）：用以表示電腦記憶體中一個位址的數字。

Apple IIe ：Apple II 系列中的一種個人電腦機型（ e 表 enhanced，坊間常稱“行家型”，其鍵盤是高級的電腦專用鍵盤。）

Apple IIe 80-Column Text Card（Apple IIe 80格文字卡）：一張週邊卡，插於Apple IIe 的輔助槽（auxiliary slot），用以將文字顯示幕由 40 格轉變成 80 格寬度。

Apple IIe Extended 80-Column Text Card（Apple IIe 擴充性 80格文字卡）：一塊週邊卡，插於Apple IIe 的輔助槽（auxiliary slot），用以將顯示幕由 40 格轉變成 80 格。同時將Apple IIe 的記憶體，擴充 64K 位元組的容量。

Applesoft BASIC ：Apple II 電腦所用 BASIC 程式語言的增訂版，能處理浮點數字。執行 Applesoft 程式的解譯程式（interpreter），建於Apple II Plus 和 Apple IIe 系統的韌體（firmwire）上。請與 Integer BASIC互相比較。

application program（應用程式）：一份程式，利用電腦的資源和

功能，來處理特定的用途或工作，例如：文字處理（word processing）、資料庫管理（data base management）、繪圖（graphics）通訊（tele-communication）。

application software（應用軟體）：電腦系統的一部份，由應用程式（application program）所組成。

argument（引數）：(1)供函數運算的值；(2)引數表示係使用於指令的值。在DOS指令中，常出現的引數有：檔名（filename fn）、槽號（slot number, Sn）、磁碟機代號（drive number, Dn）、卷號（volume number, Vn），後三個引數中的n是代表數字。

array（陣列）：一群變數，使用相同的名稱，但是以數字註標（subscript）來區分各項，以指明是一元素（element）。

ASCII：美國標準資訊交換碼（American Standard Code for Information Interchange）；一種資訊碼，以0至127的數字表示各種文數字（alphanumeric）或控制字元（control character）。ASCII碼用來表示文字，供電腦的內部使用，或電腦之間互傳資料，或者電腦與週邊設備互通訊息。

assembler（組合程式）：一種語言翻譯程式，將組合語言所寫的程式轉成對應的機器語言形式的一種程式，可視爲人看得懂的組合語言與硬體能了解的機器語言之間的"轉譯者"。

assembly language（組合語言）：一種低階的程式語言，各列機器語言指令以符號的形式寫成，使得程式比機器語言更容易爲人閱讀。

auxiliary slot（輔助槽）：Apple IIe之特殊擴充槽，供80格（80-Column）或擴充性80格（Extended 80-Column）文字卡〔Text Card〕使用。

backing up（儲備）：將檔案或磁碟中的資訊，產生一份拷貝存入其他的檔案或磁碟片。

BASIC：初學者萬用符號指令碼（Beginner's All-Purposes

Symbolic Instruction Code 之縮寫字）；一種高階的程式語言，設計得非常易學、易使用。Apple Ⅱ 電腦中有兩種BASIC版本可以使用：Applesoft（建於Apple Ⅱe 和Apple Ⅱ Plus的韌體中）與 Integer BASIC（建於Apple Ⅱ 電腦中，或由SYSTEM MASTER 磁碟片所提供）。

binary（二進位）：利用數字 0 和 1，以 2 爲底的數值表示法。使用於電腦中，因爲數值 0 或 1 很容易以各種物理狀態表示，例如：有無電流，正負電壓，顯示幕上的黑白點。

binary file（二進位機器碼檔案）：一種檔案，係將記憶體中的資訊照單存入檔案。

bit（位元）：位元化數字（0 或 1）；最小可能的資訊單位，由簡單的雙向選擇所組合而成，如：是或否，開或關，正或負，有或無。

boot（啓動）：將程式由外接儲存體，如磁碟，載入記憶體，以便啓動電腦。通常先載入一份小程式，由此小程式將更大的程式讀入記憶體。此程式好比是“pull itself up by its own bootstraps”，因以得名“bootstrapping”或“booting”。

boot disk（啓動磁碟）：一種磁碟片，存有適當安排的軟體，以便將其載入Apple Ⅱ 的記憶體中，使系統得以開始工作。通常也叫做 startup disk（開機磁碟片）。

buffer（保留區）：電腦記憶體中的一塊區域，保留做特定用途，例如：保存欲顯示於螢幕的圖形資訊，或由鍵盤和其他週邊設備讀入的文字。對於操作速度不同的設備之間傳送資訊，例如電腦的處理機與印表機或磁碟機之間，必須以這塊保留區做爲媒介，由某一設備將資訊放入此區，然後另一個速度不同的裝備由此區取得資料。（另一譯名是“緩衝區”）

bug（“虫”）：程式中的“錯誤”，通常會造成程式無法如預期的結果來工作。

byte(位元組)：由固定數量的位元所組成的資訊單位；在Apple Ⅱ電腦中，一個位元組由8個位元(bit)組成，可存任何介於0至255之間的任何數值。

call(呼叫)：表示"要求執行"一副程式(subroutine)或函數(function)的動作。

catalog(檔案目錄)：磁碟中所有檔案的一份名單；有時也稱爲檔案名錄(directory)。

character(字元)：字母、數字、標點符號、或其他用來列印或顯示資訊的可印符號。

chip(晶片)：一小片半導電性材料(通常是矽材料)，積體電路即製於其上。晶片這名詞實際上是指矽片本身，但是經常被用以指積體電路及其包裝。請參考 integreted circuit (積體電路)。

code(碼)：(1)數字或符號，用以代表離散式的資訊。(2)構成程式的指述或指令。

cold start(冷啓動)：一種啓動Apple Ⅱ的過程，首先打開電源，將作業系統載入主記憶體，然後載入一特定程式並加以執行。有特定的指令和控制字元可用來模擬冷啓動的動作。

command(指令)：使用者傳送給電腦系統的特殊訊息，指定電腦立即執行某些動作。通常由鍵盤輸入，也可以在程式中下達指令。

command file(指令檔)：可以執行其他程式的程式。EXEC指令用來執行指令檔，指令檔可包含DOS指令和BASIC指述

compiler(編譯程式)：語言翻譯程式，將高階語言程式轉換成對等的低階語言，如機器語言(machine language)，以便爾後執行。可與Interpreter (解譯程式)相互參考比較。

component(元件)：通常指電腦系統的一部份。

connector(連接器)：物理裝置，如插栓(plug)、插槽(socket

）、插座（jack），用來將系統的硬體元件接於另一元件。

control character（控制字元）：一種字元，用來控制或修改資訊的列印或顯示方式。控制字元的ASCII 碼介於0到31之間，可由鍵盤輸入；在打入英文字母的字元的同時也按住CONTROL鍵，即可輸入此類字元。例如，同時按CONTROL-M表示回到一列的開頭，相當於RETURN 。

controller card（控制界面卡）：週邊卡(peripheral card)，用以將週邊設備（peripheral device）如印表機或磁碟機，連接於Apple Ⅱ 主機，並控制此設備的動作。

copy-protect（拷貝保護）：將記錄於儲存媒體的資訊，例如存於磁碟片的商業軟體產品，以特殊的軟體（或硬體）技巧加以保護。

cursor（游標）：一個標記符號，顯示於螢幕上，標定下一個動作的發生位置，或由鍵盤輸入的下一個字元出現的位置。

data（資料）：資訊；特指使用於程式中的資訊（數據）。

data base（資料庫）：資訊的集合體，以某種形式加以架構，以供電腦系統來處理資訊。

debug（除錯）：將錯誤或出毛病的緣由，或電腦系統的故障予以挑出並修正。通常是指與軟體有關的毛病。

decimal（十進位）：日常生活中數字的表示法，數值以十爲底，利用0至9十個數字表示。與binary(二進位）和hexadecimal(十六進位）相對照。

default（預設值）：一數值或某種動作或配備，在沒有明顯指定的情況下，由電腦系統自動採用適當的數值或動作。例如，如果某一指令要執行磁碟中的程式，卻沒有指定使用那一部磁碟機，則DOS 自動採用上一個指令所使用的磁碟機。

deferred execution（延遲執行模式）：指令保存於程式中，成爲程式的一部份，以便稍後執行此程式時才執行此指令，此種情

況在指令列開頭有列號時發生。與 immediate execution（立即執行）相對照。

device（設備）：⑴物理裝置，用來執行特定工作或實現特殊的用途。⑵特指電腦系統的硬體元件。

digit（數字）：⑴ 0 至 9 任一字元，用來表示十進位數目。⑵以其他方式表示數目的字元，如二進制中的 0 和 1，十六進制中的 0 至 9 和 A 至 F。

dimension（範圍，因次）：陣列中某一註標變數的最大值。

directory（檔案名錄）：磁碟中所有檔案的名錄；也叫做檔案目錄（catalog）。

disk（磁碟）：資訊儲存媒體，由扁平圓形的磁面所構成，資訊以小磁點的形式記錄於其上，類似於聲音錄在磁帶上。

disk controller card（磁碟機控制界面卡）：週邊界面卡，可以將一部或兩部磁碟機連接在 Apple II 電腦上，並控制其操作。

disk drive（磁碟機）：週邊設備，用來將資訊讀出或寫入磁碟表面的裝置。

disk envelope（磁碟封套）：可移動的保護紙套，以便攜帶或保存磁碟片。將磁碟插入磁碟機之前必須先除去封套。與 disk jacket（磁碟護套）相對照。

diskette（磁碟片）：此術語常用來指迷你型（5 ¼吋）的軟性磁碟。

disk jacket（磁碟護套）：覆於磁碟上的永久防護套，通常是利用黑紙或黑塑膠片所製成。即使要將磁碟片插入磁碟機，也不必除去護套。

Disk Operating System（DOS，磁碟作業系統）：Apple II 所用的軟體系統，使電腦得以控制一或數部磁碟機，並且與其相互溝通。

disk-resident（常駐磁碟的）：經常性地儲存或保留於磁碟上。

display（顯示）：(1)展示眼前的資訊，尤其指展示於顯示裝置的畫面上。(2)展示資訊於眼前。(3)顯示裝置。

display device（顯示設備）：顯示可見資訊的設備，如電視機或影像監視器（video monitor）。

display screen（顯示幕）：玻璃或塑膠面版，座落於顯示裝置的前端，以顯示圖像。

DOS: 參考Disk Operating System（磁碟作業系統）。

drive（磁碟機）：參考disk drive。

80-column text card（80格文字界面卡）：週邊界面卡，插於Apple Ⅱ 的擴充槽或Apple Ⅱe 的輔助槽（auxilliary slot），用來將電腦的文字顯示畫面由40格文字轉變成80格。

element（元素）：集合（set）或集合體（collection）的成員；特指組成陣列的一單獨變數。

error message（錯誤訊息）：顯示或列印出來的訊息，指出程式執行中所發生的錯誤或問題。

EXEC file：利用EXEC指令執行的程式檔，此種程式會帶出其他程式，並加以執行。

execute（執行）：實施或完成一項特定的行動或一系列行動，如程式所描述的動作。

expansion slot（擴充槽）：Apple Ⅱ 電腦中的連接器，用來裝設週邊介面卡；也叫週邊槽（peripheral slot）。

field（資料欄）：一串1至32767個字元不等的字串，以歸位（carriage-return）字元作結束。

file（檔案）：資訊的集合體，以檔名識別，存入如磁碟之類的週邊儲存媒體。

filename（檔名）：儲存檔案時所用的名稱，以資識別。

file type（檔案型態）：佔一個字元的編碼，用以識別檔案的內容型式，指示以何種方式使用檔案。

firmware （韌體）：電腦系統中的元件，由程式所組成，永久儲存於唯讀記憶體（read-only memory）中。此種程式（例如，Applesoft 解譯程式和監督程式），在工廠生產線製造時就建入電腦中；可以隨時執行，但是無法在記憶體中將其消除或修改。與 hardware（硬體）和 software（軟體）相對照比較。

fixed-point （定點）：電腦內部表示數目的一種方法，此法中小數點係出現於數字的固定位置。標準做法是將小數點置於數字的最右邊，使得數字可視為“整數”。相同長度表示數字時，定點數字所涵蓋的範圍比浮點數字小，但是準確度較大。

flexible disk （軟性磁碟）：用軟性塑膠製成的磁碟，通常叫做 floppy disk 。與 rigid disk（剛性磁碟）相對照。

floating-point （浮點）：電腦內部中一種數目表示法，小數點可在數字中的任何位置「浮」動。數字中有數個位元記錄小數點的位置。固定長度的情況下，以浮點表示數目所能涵蓋的範圍比定點數目要大，但是準確度較小。與 fixed-point 相對照。

floppy disk （軟性磁碟）：參考 flexible disk 。

format（格式）：(1)資訊組織或展示的格式。(2)指定或控制資訊的格式。(3)準備空白磁碟以供存放資訊，此動作將磁碟表面劃分出磁軌和磁區；參考 initializing （啓始化）。

function（函數）：已寫好的計算程式，可在程式中任何位置叫用。

hardware（硬體）：電腦系統中的元件，由物理裝置（電子或機械的）所組成。與 software（軟體）和 firmware 韌體）相對照

hexadecimal（十六進位）：以十六為底的數字表示法，利用 0 至 9 和 A 至 F 來表示。十六進位表示數字比二進位數字易讀且容易理解，同時又能很容易轉換成二進制：每一位十六進位數字正好對應 4 個二進位數字（亦即 4 個位元相當於一個十六進位的表示法）。

high-level language （高階語言）：“人們”比較容易理解的程

式語言。通常高階語言中的一條指述，會對應機器語言的數條指令。BASIC 即屬於高階語言。

high-order byte （高位元組）：記憶體位址或其他雙位元組數目（16 位元表示）中較高位的八位元所在的位置。在 Apple II 的 6502 微處理機中，位址的低位元組存於先，高位元組存於後。

IC：參考 integrated circuit。

immediate execution （立即執行模式）：程式列的執行時機在輸入此列時立即發生，亦即是在輸入"沒有列號"的程式列即能引導電腦作某事。與 deferred execution （延遲執行模式）相對照。

indexed addressing（索引定址法）：指定記憶位址的一種方法，用於機器語言程式。

information （資訊）：以有組織的方式來表示的事實、觀念或指示。經常是意味著"程式"、"數據"、或"資料"。

initialize （啓始）：(1)設定起始狀態或起始值，以作爲計算作業的準備。(2)準備空白磁碟，以便接收資訊，此動作是將磁碟表面劃分出數十個磁軌和磁區，此項工作又稱 formatting（規劃格式）。

initialized disk （已啓始化的磁碟）：已規劃好磁軌和磁區的磁碟，並存有 DOS 和歡迎詞程式（greeting program）此DOS所用的記憶體位址，與準備此磁碟的電腦之 DOS 相同。INIT 指令即是用來建立一片可用的磁碟。

input（輸入）：(1)由外接設備，如鍵盤或磁碟機，傳入電腦的資訊。(2)傳送此類資訊的動作程序。

instruction（機器指令）：機器語言或組合語言程式的基本單位，對應於電腦處理機的單一動作。

integer （整數）：整個數目，沒有分數部份，在電腦內部以定點（fixed-point）形式表示。

integer BASIC ：BASIC 程式語言的一種版本，使用於 Apple II 電腦。Integer BASIC 比Applesoft 老，只能處理整數（定點 fixed-point）。SYSTEM MASTER 磁碟片存有一份解譯程式，可用來建立和執行 Integer BASIC 所寫的程式。當利用此磁碟片啓動電腦時，會自動將其載入記憶體。

integrated circuit （積體電路）：電子元件，由很多電路元素所組成，這些電路製入一小片半導體材料。請參考 chip（晶片）。

interface （界面）：設備、規則、習慣，使得系統的某一部份得以和另一部份溝通。

interface card （界面卡）：週邊界面卡，用來解決某一特殊界面的訊號傳輸問題，使電腦得以和週邊設備溝通，如印表機。

Interpreter (解譯程式）：一種語言翻譯程式，將某種程式語言寫成的程式一列一列依次讀入，每讀入一列指述立即執行其所指定的動作。坊間有人稱之爲"直譯程式"。與 compiler（編譯程式）相對照。

Interrupt（中斷）：暫停執行程式，使電腦可以從事其他的工作，此種工作通常是回應週邊設備傳來的訊號或者其他外接裝置的訊號。

I/O（輸入／輸出）：係 input／output 的縮寫。此術語常用來指與電腦相溝通的設備與程序，或相互溝通的資訊。

I/O device （輸出入設備）：即 input／output device，用來傳送資訊出入電腦。參考 input （輸入）、output（輸出）、peripheral device（週邊設備）。

I/O link （輸出入鏈）：某一固定的位址，存有Apple II 監督程式的輸出入副程式之位址。

K（仟）： 2的 10 次方或1024 （由希臘字根 Kilo取其K表示一仟）。例如，64 K相當於 64×1024 或 65,536。

Keyboard(鍵盤）：建於 Apple II 電腦上的字鍵面盤，類似於打字

機的鍵盤，在其上輸入資訊給電腦之用。

Kilobyte（仟位元組）：由 1 K（1024）位元組所組成的資訊單位，或相當於 8 K（8192）位元。參考K（仟）。

KDW：Apple Ⅱ 電腦中某一位址的符號，此位址存標準輸入鏈。此符號係Keyboard switch（鍵盤開關）的縮寫。請參考 I／O link（輸出入鏈）。

language（語言）：參考 programming language（程式語言）。

language translator（語言翻譯程式）：系統程式之一種，其工作是讀入某種以特定程式語言所寫的程式，直接執行此程式，或者將其轉換成別種語言（如機器語言）以便爾後執行。

load（載入）：將資訊由週邊儲存媒體（如磁碟）轉存至主記憶體以供使用。例如，將程式轉入記憶體以便執行。

loader（載入程式）：將BASIC或二進位機器碼檔案由磁碟載入主記憶體的程式。

lock（鎖定）：限制檔案的使用方式，使其不致於被修改、刪除、改名。鎖住的檔案在DOS 的檔案目錄（catalog）中標有星號（*）。與unlock（解鎖）相對照。

low-level language（低階語言）：程式語言中之一類，比較接近電腦的 CPU 處理機可以執行的一種形式之語言。

low-order byte（低位元組）：記憶位址或其他雙位元組數目（以 16 位元表示）中較低位的 8 位元。在Apple Ⅱ 的 6502 微處理機中，位址的低位元組存於先，高位元組存於後。

machine language（機器語言）：電腦指令的形式，可直接存入記憶體中由處理機來執行。各種類型的電腦處理機（如Apple Ⅱ 電腦使用 6502 微處理機）都有各自的機器語言。

main memory（主記憶體）：電腦系統的記憶元件，建入電腦內部，處理機可直接存取其內容。

master disk（主磁碟）：存有DOS 的磁碟片，而且此 DOS 係

儘可能存於最高記憶位址的版本。因此，這種磁碟可在任何 16K 或更大的系統之下操作。利用MASTER 程式可由已啓始磁碟轉換成主磁碟。

memory（記憶體）：電腦系統的硬體元件，可以存放資訊供爾後使用。參考main memory （主記憶體）、randon-access memory （隨意存取記憶體）、read-only memory(唯讀記憶體）。

memory location（記憶體位址）：主記憶體的基本單位，利用位址（address)來識別，可以保存單項固定長度的資訊。在 Apple II 中，記憶體位址可保存一個位元組的資訊。

memory-resident（常駐記憶的）：(1)永久存在主記憶體，如韌體（firmware）。(2)持續存在主記憶體，即使不使用亦然，如磁碟作業系統（DOS）。

menu（功能表）：一份選擇清單，由某一程式展現出來，通常出現在螢光幕上，讓使用者挑選。

microcomputer（微電腦）：電腦之一類型，如 Apple II 即屬此類，其處理機採用微處理機者。

microprocessor（微處理機）：電腦之處理機容入單一積體電路晶片中，如使用於 Apple II 的 6502 微處理機。

mode(模式）：電腦或系統的狀態，做爲決定其行爲之依據。

modem(數據機）：調變／解調器（modulator／demodulator）之縮寫，一種週邊設備，使電腦得以經由電話線傳送或接收資訊。

monitor（監視器）：參考video monitor （視訊監視器）。

Monitor program（監督程式）：建於 Apple II 電腦內部之一份系統程式，可用來直接檢查或修改記憶體之內容，並以機器語言來操作電腦。

nibble（半位元組）：資訊的單位相當於半個位元組，或 4 個位元，可存任何 0 至 15 的數值。有時拼做 nybble。

operating system（作業系統）：軟體系統，用來管理電腦的資源和功能，使其可供使用者或電腦執行的應用程式利用。參考 Disk Operating System（磁碟作業系統）。

option（選擇項）：可省略的引數（argument），或稱"可有可無"的一種選擇性引數。

output（輸出）：由電腦傳到外接裝置的資訊，如顯示幕、磁碟機、印表機。(2)傳送此類資訊的程序。

Pascal：一種高階程式語言。有一種版本稱做Apple Pascal，可使用於Apple II 系列的電腦。

peripheral（週邊的）：位於電腦外圍的，可以是指實體裝置（如週邊設備）或也可以表示邏輯意義的（如週邊界面卡）。

peripheral card（週邊界面卡）：活動式的印刷電路版，插於Apple II 的擴充槽，可用以連接週邊設備或執行某些附屬週邊的功能，以擴充或更改電腦的能力。

peripheral device（週邊設備）：由電腦所控制的輔助裝置，此類例子有印表機（printer）、錄音機（cassette recorder）磁碟機（disk drive）。

peripheral slot（週邊槽）：參考 expansion slot（擴充槽）。

pointer（指標）：一種資訊項目，存有某些其他項目所在的記憶體位址值。例如，Applesoft 保存一份內部指標，指向最近使用的內儲變數，最近輸入的程式列和最近讀入的資料項目。

printed-circuit board（印刷電路版）：電腦或其他電子設備的硬體元件，由平面四方的剛性材料所製成，通常是玻璃纖維，積體電路和電子元件即連於其上。

printer（印表機）：一種週邊設備，用來將資訊寫於紙上。

processor（處理機）：電腦的硬體元件，直接執行主記憶體中的機器語言指令，以從事眞正的計算和控制工作。

program（程式）：一組指令，依照某一特定程式語言的規則和習

慣寫成，指示電腦動作以完成某些工作。

programming language（程式語言）：用來寫程式的一組規則和慣用法。

prompt（提示）：提醒或指示使用者，電腦期待某些動作，通常顯示一個特殊的符號、醒目的訊息、功能選擇表於螢幕上。

prompt character（提示字元）：顯示於螢幕上的特殊字元，用來提醒使用者下一步動作；通常也用來識別提供此項提示的系統程式或元件。例如，提示字元右方括弧記號] 用於 Applesoft BASIC 解譯程式；大於記號＞用於 Integer BASIC；星號 * 用於系統監督程式。也稱做 prompting character 。

radio-frequency modulator（射頻調變器）：一種設備，用來將電腦的視訊轉換成電視可以接受的訊號。

RAM（隨意存取記憶體）：參考 random-access memory。

random-access memory（隨意存取記憶體）：一種記憶體，其各位址的內容可以不按任何順序予以直接存取。這個名詞常用來指讀寫記憶體（read-write memory）。因爲嚴格說，無論是唯讀（read-only）或讀寫記憶體都可以不按順序隨意存取。請與 read-only memory（唯讀記憶體）和 read-write memory（讀寫記憶體）相對照比較。

read（讀取）：由電腦的外接來源（如磁碟機）將資訊轉入電腦的記憶體，或者由外接處理機（如鍵盤或主記憶體）傳入電腦的處理機之動作。

read-only memory（唯讀記憶體）：一種記憶體，僅能讀取其內容，但不能再寫入資料。資訊只有在生產製造時才寫入一次，然後永久保存於記憶體中。即使電腦的電源關掉，其內容也不會消失或更動。與 read-write memory（讀寫記憶體）和 random-access memory（隨意存取記憶體）相對照。

read-write memory（讀寫記憶體）：一種記憶體，其內容可寫

也可讀，通常誤叫做 random-access memory(隨意存取記憶體。當電腦的電源關掉時，存於讀寫記憶體中的資訊永遠消失，除非將其儲存到可永久儲存的媒體，如磁碟。與 read-only memory（唯讀記憶體）和 random-access memory(隨意存取記憶體）相對照。

register（暫存器）：電腦處理機中的位址，可在程式的控制之下保存或修改資訊。

resident（常駐）：參考 memory-resident（常駐記憶體）、disk-resident（常駐磁碟的）。

rigid disk（剛性磁碟）：利用硬性而無彈性的材料所製成的磁碟機，也稱做 hard disk（硬性磁碟）。與 flexible disk(軟性磁碟）相對照。

ROM(唯讀記憶體）：參考 read-only memory。

run（執行）：(1)執行一份程式。(2)將程式由週邊儲存媒體如磁碟，載入主記憶體，並加以執行。

save（儲存）：將資訊由主記憶體傳送到週邊儲存媒體，以備爾後的使用。

screen（螢幕）：參考 display screen（顯示幕）。

scroll（捲動）：改變全部或部份顯示幕的內容，其方法是將資訊往畫面的某一端移動（通常是上端），使得另一端讓出空間顯示新的資訊（通常是下端一列）。其產生的效果，好比移動卷軸通過固定的窗口一般。

sector（磁區）：磁碟表面之一片段，由某一磁軌（track)中固定長度的某一片段所組成。DOS 3.3 中，每一磁軌有 16 磁區。

silicon（矽）：非金屬、半導電性化學元素，是爲積體電路之原料。與 silica（矽氧化物）如石英（quartz)、貓眼石（opal）、沙質材料，或者 silicone（任何含矽的有機混合物），有所區別。

slot（槽）：參考 expansion slot（擴充槽）。

software（軟體）：電腦系統之元件，由程式所構成，可用以決定或控制電腦的行為。與 hardware（硬體）和 firmware（韌體）相對照。

space character（空白字元）：一文字字元，可顯示成空白，由鍵盤按下空白鍵 SPACE，即可輸入此字元。

startup disk（啟動用磁碟）：一種磁碟片，存有適當形式的軟體，可載入 Apple II 的記憶體，使系統得以進入工作狀態。有時也稱做 boot disk。

statement（指述）：高階語言程式中的基本單位，指定某一特定動作，由電腦來執行，通常一指述可對應機器語言的數個指令。

subroutine（副程式）：程式之一部份，可在程式中任一點呼叫執行某一副程式，執行完此副程式之後，控制權又回到呼叫點。

subscript（註標）：索引號碼（index number），用來指定陣列之某一元素。

syntax（語法）：程式語言中主宰指述或指令結構的規則。

system（系統）：相關且相互作用的數部份組織成的有協調之集合體，以便執行某些功能或實現某些目的。

system program（系統程式）：程式之一種，可提供電腦的資源和功能給一般性的用途，如作業系統或語言翻譯程式。與 application program（應用程式）相對照。

system software（系統軟體）：電腦系統之元件，由系統程式所組成

telecommunications（電傳通訊）：長距離的資訊傳輸，如通過電話線和數據機（modem）達成不同用戶和設備之間資訊的流通。

television set（電視機）：可利用天線接收廣播視訊的顯示設備（如商業電視）。可與射頻調變器（radio frequency modulator）組合，而成為 Apple II 電腦的顯示設備。與 video mo-

nitor（影像監視器）相對照。

text（文字）：⑴資訊以字元形式來表示，以供人來閱讀。⑵顯示於 Apple Ⅱ 螢幕上的字元（可能是多個字元構成的文句）。

text file（文字檔）：一種檔案，以文字的形式儲存資訊。

text window（文字窗）：Apple Ⅱ 的顯示幕上之一塊區域，在此區域內可顯示或捲動（scroll）文字。

track（磁軌）：磁碟表面的一塊區域，由磁碟上的某一同心環帶所組成。在DOS 3.3 的磁碟上有 35 個磁軌（track）。

turnkey disk（轉鑰式磁碟）：一種磁碟，在磁碟啓動時可執行某一特定程式。

turnkey program（轉鑰式程式）：一種程式，可能是遊戲程式或應用程式，當磁碟啓動時會自動執行此程式。

unlock（解鎖）：移去某些使用檔案的限制，使得此檔案可以再修改、重命名、或刪除。與 lock（鎖定）相對照。

variable（變數）：⑴電腦記憶體中之位址，可用以存值。⑵程式中的符號，可用來表示此類位址。

video monitor（影像監視器）：一種顯示設備，只能經由直接連線來接收視訊，而無法如商業電視接受廣播訊號。可直接連於 Apple Ⅱ 電腦上做爲顯示設備。

volume（卷册）：存放於磁碟上的資訊集合體。在Apple 系統上的一片磁碟即相當於"一卷"資訊（類似書籍的"第幾卷"慣用法）。

warm start（溫啓動）：在電源已開啓之後，又重新啓動 Apple Ⅱ 的程序，此動作不會將操作系統重新載入主記憶體，而且通常不至於失去記憶體中的程式與資訊。

wildcard character（通配字元）：等號＝，可表示檔案名錄中的檔案或檔名字元部份相同之檔案。

write（寫入）：將資訊由電腦傳到外接設備，如磁碟機或印表機。

write-enable notch （可寫缺口）：磁碟護套邊緣的小方缺口，使得資訊得以寫入磁碟，如果沒有可寫缺口或缺口已蓋上貼紙，則僅可以由磁碟讀出資料，但是不能寫入資料。

write-protect （寫入保護）：保護磁碟中的資訊，可在可寫缺口（write-enable notch）貼上"寫入保護"貼紙（write-protect tab），以防止任何新的資訊寫入磁碟。

write-protect tab （寫入保護貼紙）：一小塊黏性貼紙，可貼於可寫缺口（write-enable notch），以防資訊寫入磁碟。

附錄 快速提示表

啓始化一磁碟

1. 啓動DOS；將空白磁碟片放入磁碟機。
2. 打入一份歡迎詞程式，類似下列的例子：

```
10 PRINT "32K DISK INITIALIZED
    5 MAY 84"
20 END
```

3. 將此程式命名爲HELLO，並立即進行規劃磁碟格式的動作，打入

```
INIT HELLO
```

建立一主磁碟

1. 在已啓始化的磁碟上，建立一新的歡迎詞程式，並以新檔名儲存於磁碟上。
2. 將 SYSTEM MASTER 磁碟插入磁碟機內，並打入

```
RUN MASTER
```

3. 鍵入第 1 步驟所建立的歡迎詞程式之檔名；若想使此磁碟片成爲轉鑰式磁碟，請先按下 [ESC] 鍵再鍵入檔名。
4. 抽換 SYSTEM MASTER 磁碟片，改插入準備轉換的已啓始化磁碟。按下 [RETURN] 鍵即能開始進行轉換成主磁碟的動作。

引數

fn　檔名，1 至 30 個字元（除了逗點之外的任何可印出的文字），必須以字母起頭。

n　代表您指定的十進制或十六進制數字（用在PR#，IN#和MAXFILES等指令）。若是十六進制表示法，須以錢號（$）再接著十六進制數字，例如$FE。

An　指明記憶體內的某一位址，其範圍介於 0 至 65535之間。

Bn　指明一“絕對”的位元組編號，其範圍介於 0 至 32767之間。若略去本項，DOS預設其值爲 0 。

Dn　指明適用的磁碟機（代號 1 或 2 ）。

Ln　指明位元組的長度，其範圍介於 0 至 32767之間。在OPEN指令中使用時，它代表“資料錄長度”（隨意存取文字檔）。用在BSAVE 指令中時，它代表自記憶體位址 An 起算共有多少位元組。若略去本項，DOS預設其值爲 0 。

Rn　指明一“相對”的位元組編號，其範圍介於 0 至 32767之間。在OPEN指令下達後，它預設成 0 ，亦即是相當於檔案的第一個資料錄。而後，它設定成最近存取過的資料錄編號。

Sn　指明適用的插充槽（編號 0 至 7 之間）。

Vn　指明適用的卷號，其編號介於 0 至 254 之間。

指令摘要

DOS 的指令可細分成下列五類：

存取磁碟的非程式化指令：

CATALOG	RENAME	VERIFY	SAVE
INIT	LOCK	RUN	PR#
DELETE	UNLOCK	LOAD	IN#

控制程式環境的指令：

FP	MON	EXEC	CHAIN
INT	NOMON	MAXFILES	

循序存取的指令：

OPEN	READ	APPEND
CLOSE	WRITE	POSITION

隨意存取的指令：

OPEN	CLOSE	READ	WRITE

二進制機器碼檔案的指令：

BRUN	BLOAD	BSAVE

控制程式環境的指令

Fp [, Sn] [, Dn] [, Vn]

將電腦切換成Applesoft 語言系統，會清除先前已存於記憶體的指標值及程式內容。

INT

將電腦切換成 Integer **BASIC** 語言系統，會清除先前已存於記憶體的指標及程式內容。

MON [C] [, I] [, O]

監督程式的執行流程；顯示指令名稱（ C 選擇項），自磁碟讀入的資料（ I 選擇項），和輸出至磁碟的資料（ O 選擇項）。

NOMON [C] [, I] [, O]

抑制監督流程的顯示；指令名稱（ C 項），自磁碟讀入的資料（ I 項）和輸出至磁碟的資料（ O 項）。

EXEC fn [, Rn] [, Sn] [, Dn] [, Vn]

（批次）執行先前以 BASIC 程式建立的 DOS 指令檔。可指定自檔案的第 R 筆資料錄開始執行。

MAXFILES n

指明在同一時間內可同時開啓的檔案個數。n 值介於 1 至 16 之間。僅可在執行一程式之前下達本指令。

CHAIN fn [, Sn] [, Dn] [, Vn]

執行指名的 Integer BASIC 程式；它不會關閉目前已開啓的檔案，也不清除既存於記憶變數的內含值。

存取磁碟的指令

CATALOG [, Sn] [, Dn]

顯示磁碟的目錄，包含磁片卷號，磁碟內所有檔案之名稱，檔案型態及佔用的磁區個數，以及那些檔案已鎖定保護。

INIT fn [, Sn] [, Dn] [, Vn]

規劃磁碟的格式，它可將 DOS 本身及一歡迎詞程式存入磁碟內。

DELETE fn [, Sn] [, Dn] [, Vn]

自磁碟中刪除指名的檔案。

RENAME fn1, fn2 [, Sn] [, Dn] [, Vn]

將既存於磁碟的檔案，其原檔名 fn1 更改成新檔名 fn2。

LOCK fn [, Sn] [, Dn] [, Vn]

鎖定一檔案（以預防它無意中被刪除或寫入資料更改其內容）。

UNLOCK fn [, Sn] [, Dn] [, Vn]

解除已鎖定檔案的保護措施。

VERIFY fn [, Sn] [, Dn] [, Vn]

測試指名的檔案是否能爲 DOS 自磁碟中讀取並存入記憶體。

RUN fn [, Sn] [, Dn] [, Vn]

將指名的 BASIC 程式自磁碟中讀取出來，拷貝一份存入記憶體內，並立即執行它。

LOAD fn [, Sn] [, Dn] [, Vn]

將磁碟中的程式，讀取其內容存入記憶體，準備執行它。

SAVE fn [, Sn] [, Dn] [, Vn]

將目前已存於主記憶體的 BASIC 程式，存入磁碟中產生一個檔案，其檔案型態是A或I（依據目前使用中的BASIC語言型態）。

PR # n

將原本送至螢幕顯示的字元，改道傳送至連接於編號 n 的擴充之裝置上。

IN # n

本指令下達後，往後即自連接於編號 n 的擴充槽之裝置上取得輸入值。

循序存取的指令

循序存取的意義：當開啓一檔案時不必指明它的資料錄長度，其內各錄需依序地一筆一筆讀取或寫入。每次檔案開啓時，即自檔案的起始端讀取或寫入資料。檔案必須是文字檔（其型態是T）。

OPEN fn [, Sn] [, Dn] [, Vn]

開啓一循序存取的文字檔，並分配給此檔案一個“保留區”。假若此檔尚未存在於磁碟中，DOS負責建立此檔；若此檔已存在，

則不可重複開啓它（除非先關閉它）。最多可有 16 個檔案同時被開啓。

CLOSE [fn]

完成寫入檔案的動作並關閉指名的檔案（若未指定檔名，則關閉除執行中的指令檔之外的所有檔案）。釋回各相關檔案所用的保留區。

READ fn [, Bn]

指明被讀取（INPUT 或GET指令）的循序存取文字檔；若加註 Bn 引數，即是指定自檔案讀取的起始處，指名的檔案必須已開啓（OPEN 指令）。

WRITE fn [, Bn]

指明被寫入資料（PRINT 指令，或系統產生的錯誤訊息以及任何提示符號）的循序存取文字檔，以及自檔案那個位置開始寫入之。指名的檔案必須已開啓。

APPEND fn [, Sn] [, Dn] [, Vn]

開啓一循序存取的文字檔，並且準備自檔案的結尾處開始寫入（WRITE 指令）。指名的檔案必須是已存在的檔案。

POSITION fn [, Rn]

設定下次讀取或寫入資料的目前檔案指標於相距 Rn 個歸位字元之處。

錯誤代碼及訊息

代碼	訊息
1	LANGUAGE NOT AVAILABLE
2,3	RANGE ERROR
4	WRITE PROTECTED
5	END OF DATA
6	FILE NOT FOUND
7	VOLUME MISMATCH
8	I/O ERROR
9	DISK FULL
10	FILE LOCKED
11	SYNTAX ERROR
12	NO BUFFERS AVAILABLE
13	FILE TYPE MISMATCH
14	PROGRAM TOO LARGE
15	NOT DIRECT COMMAND

隨意存取的指令

隨意存取的意義：開啓一檔案時必須同時指明它的資料錄長度。指明的 fn 檔案必須是T 型態的文字檔。DOS可在您指定編號的資料錄存取其資料，而不必顧及上一次是在那個資料錄讀取或寫入資料。

OPEN fn, Ln [, Sn] [, Dn] [, Vn]

開啓一隨意存取的文字檔。分配此檔案一個保留區。假若此檔尚未存在於磁碟中，則 DOS 建立此檔；若已存在，則不可再次開啓它（除非先關閉它）。Ln 引數即指明資料錄的固定長度。最多可同時開啓 16 個檔案。

CLOSE [fn]

完成寫入檔案的動作並關閉指名的檔案（或者除指令檔之外的所有已開啓的檔案）；釋回相關的各檔案之保留區。

READ fn [, Rn] [, Bn]

指明被讀取（INPUT 或GET 指令）的隨意存取文字檔，以及

自何處開始讀取資料。R項引數中的n即是資料錄編號（檔案開啓時已指定是L位元組長度），B項引數中的n值即是在資料錄的位元組編號。指名的檔案必須已開啓。

WRITE fn [, Rn] [, Bn]

指明被寫入資料（PRINT指令或系統產生的錯誤訊息以及任何提示符號）的隨意存取文字檔，以及自檔案那個位置開始寫入之。R項引數的n值是資料錄編號（檔案開啓時已指定是L位元組長度），B項引數中的n值是在資料錄的位元組編號。指名的檔案必須已開啓。

二進制機器碼檔案適用的指令

BRUN fn [, An] [, Sn] [, Dn] [, Vn]

將指名的二進制機器碼檔案（B型態）自磁碟中讀取出來，存入記憶體中的一段區域（自位址 An 存起），並立即執行它。

BLOAD fn [, An] [, Sn] [, Dn] [, Vn]

將指名的二進制機器碼檔案（B型態）自磁碟中讀取出來，存入記憶體中的一段區域（自位址 An 存起）。

BSAVE fn, An, Ln [, Sn] [, Dn] [, Vn]

將記憶體中自位址 An 起共有Ln 位元組長度的二進制資料存入磁碟中，建立 fn 檔案（其型態是B）。若 fn 檔案尚未存在於磁碟中，則 DOS 建立此檔；若已存在，則直接覆蓋原先的內容。

Apple DOS 3.3磁碟作業系統手册

譯　者：詹朝基　蔡源斌

出版社：聯星圖書公司

名　行：聯星圖書公司

地　址：香港洛克道168地下

印　刷：美華印刷廠

地　址：香港柴灣工業大厦二樓128號

聯星出版社　印行

H.K. $45.00

Apple DOS 3.3 磁碟作業系統手册

詹朝基
蔡源斌 譯

聯星